European Journal of Operational Research 312 (2024) 978–995

Contents lists available at ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/ejor

Production, Manufacturing, Transportation and Logistics

A branch-cut-and-price algorithm for the time-dependent electric vehicle routing problem with time windows

Gonzalo Lera-Romero[a,b], Juan José Miranda Bront[c,d,*], Francisco J. Soulignac[a,b]

[a] Universidad de Buenos Aires, Facultad de Ciencias Exactas y Naturales, Departamento de Computación, Ciudad de Buenos Aires, C1428EGA, Argentina
[b] CONICET-Universidad de Buenos Aires, Instituto de Investigación en Ciencias de la Computación (ICC), Ciudad de Buenos Aires, C1428EGA, Argentina
[c] Universidad Torcuato Di Tella, Ciudad de Buenos Aires, C1428BCW, Argentina
[d] Consejo Nacional de Investigaciones Científicas y Técnicas, Argentina

ARTICLE INFO

Article history:
Received 24 May 2022
Accepted 21 June 2023
Available online 14 July 2023

Keywords:
Routing
Electric vehicle routing problem
Time-dependent times
Branch cut and price
Labeling algorithms

ABSTRACT

The adoption of electric vehicles (EVs) within last-mile deliveries is considered one of the key transformations towards more sustainable logistics. The inclusion of EVs introduces new operational constraints to the models such as a restricted driving range and the possibility to perform recharges en route. The discharge of the typical batteries is complex and depends on several variables, including the vehicle travel speed, but most of the approaches assume that the energy consumption depends only on the distance traveled. This becomes relevant in different logistics contexts, such as last-mile distrubtion in large cities and mid-haul logistics in retail, where traffic congestion affects severely the travel speeds. In this paper, we introduce a general version of the Time-Dependent Electric Vehicle Routing Problem with Time Windows (TDEVRPTW), which incorporates the time-dependent nature of the transportation network both in terms of travel times and the energy consumption. We propose a unifying framework to integrate other critical variable times arising during the operations previously studied in the literature, such as the time-dependent waiting times and non-linear charging times. We propose a state of the art branch-cut-and-price (BCP) algorithm. Based on extensive computational experiments, we show that the approach is very effective solving instances with up to 100 customers with different time dependent configurations. From a managerial standpoint, our experiments indicate that neglecting the travel speeds can affect the quality of the solutions obtained, where up to 40 percent of the infeasibilities induced by neglecting the time dependency can be caused by exceeding the battery capacity.

1. Introduction

One of the key environmental challenges nowadays relies in the use of cleaner and more sustainable energy resources to reduce emissions and pollution worldwide. Providing access to new and sustainable transportation solutions appears as one of the urgent topics to be addressed from a logistics perspective. As of 2010, it is estimated that 20% of the emission of greenhouse gas (GHG) in that region stemmed from transportation activities.

In the last decade, many companies have been shifting towards more environmentally friendly transportation alternatives. One area of significant impact is last-mile logistics, where traditional internal combustion engine vehicles (ICEVs) are being replaced by electric vehicles (EVs) and cargo bikes, especially in highly congested areas. For instance, UPS aims to have 25% of its vehicles running on alternative fuel by 2020, including 10,000 EVs (UPS, 2019). From a technical perspective, EVs offer several benefits compared to the ICEVs, such as less noise contamination, reduced CO_2 emissions, and a high energy conversion efficiency. They are usually combined with other methods such as the regenerative braking system, which produces electrical energy from movement. As a counterpart, the restricted battery capacity influences the operations of EVs by reducing the so-called driving range. Then, routing plans for EVs should incorporate these new operational constraints explicitly, preventing vehicles from running out of battery during the distribution by making intermediate stops at recharging stations. Recent advances have increased the driving range in the last few years, and current implementations suggest it is reaching reasonable levels for distribution within cities. However, the battery capacity declines over time and their continuous replacements represent significant investments. Other aspects such as the use of air conditioning or heating systems also reduce the driving

* Corresponding author at: Universidad Torcuato Di Tella, Av. Figueroa Alcorta 7350, C1428BCW, Ciudad de Buenos Aires, Argentina.
E-mail addresses: gleraromero@dc.uba.ar (G. Lera-Romero), jmiranda@utdt.edu (J.J. Miranda Bront), fsoulign@dc.uba.ar (F.J. Soulignac).

https://doi.org/10.1016/j.ejor.2023.06.037
0377-2217/

range, which can be up to 30% of the original capacity according to Restrepo et al. (2014).

The Electric Vehicle Routing Problem (EVRP) introduced by Schneider et al. (2014) extends the classical Vehicle Routing Problem (VRP) by incorporating explicitly the battery of the vehicle. It is modeled as an additional resource where the energy is consumed when the vehicle moves along the network and can be recharged en route. Most of the literature considers a simplified model where the battery discharge is proportional to the distance traveled, assuming the consumption rage remains constant during the planning horizon. A similar observation holds for the charging times at the fueling stations. In practice, according to Goeke & Schneider (2015) and as observed in Schneider et al. (2014) the vehicle load, the travel speed, and the gradient of the terrain are among the most important variables affecting the energy consumption.

These parameters become relevant in practice when implementing the routing plans. For instance, within last-mile logistics in large cities, variations traffic variations impact directly on the travel speeds at different moments of the day, which in turn affect the battery discharge. In this context, the packages to be distributed are usually rather small and the load is not as important as the timing decisions. If the driving range represents a limitation, either because of the available technology or degradation of the battery, using the distance as a proxy for energy consumption may be misleading. Another relevant area regards mid-haul logistics, where the transition to an electrified fleet includes some additional challenges compared to last-mile operations, as discussed in Schiffer et al. (2021). A standard mid-haul route naturally exceeds the driving range of nowadays EVs, making the operations more complex as en-route recharges may not be avoided. Congestion effects also play an important role in this case as, depending on the geographical location of the warehouses, distribution centers, and stores, the travel times and speeds can be heavily affected during rush hours, especially if considering large urban areas. We acknowledge that the total load of the truck, and how this impacts the operations, would be also an important factor to evaluate, not included as part of our research. Overall, these two scenarios indicate that en route battery levels could be wrongly estimated under the classical linear discharge model, leading to tactical plans that may not be applicable in practice due to battery depletion.

Although congestion is a dynamic phenomenon, where unexpected events may be difficult to predict in day-ahead planning, designing more realistic distribution plans is still a relevant area of study. Routing plans for next-day deliveries that are aware of the average congestion patterns are expected to translate into improved operations, possibly with fewer disruptions due to predictable traffic variations. Our paper contributes in this direction. We study the time-dependent EVRPTW (TDEVRPTW), which incorporates the effects of congestion into the EVRPTW originally proposed in Schneider et al. (2014), not only in terms of the timing but also on the charge of the battery. Furthermore, we provide a general framework that naturally integrates other relevant operational aspects affecting the applicability of the routing plans, such as variable charging times and waiting times at the recharge stations. We first provide a discussion on the literature related to the EVRPTW and the TDVRP to frame our research, and then we outline our main contributions.

1.1. Literature review

As described in the next section, our paper contributes by introducing the TDEVRPTW both in terms of the modeling as well as devising a state-of-the-art Branch-Cut-and-Price (BCP) algorithm as a solution methodology. Regarding exact algorithms, BCPs are the most effective algorithms, especially for multi-vehicle VRPs. We refer the reader to Toth & Vigo (2014) for a detailed introduction to BCP algorithms applied to VRPs, and further explore Irnich & Desaulniers (2005) and Irnich (2008) regarding advanced labeling algorithms for shortest path problems with resource constraints, including the use of Resource Extension Functions (REFs). More recently, Costa et al. (2019) provide a complete overview of the different components that are usually incorporated in a BCP.

Environmental aspects within VRPs have been addressed from different perspectives. A stream of research related to *green logistics* considers, among others, the effects of transportation on the environment by minimizing fuel consumption and GHG emissions under different congestion conditions (see, e.g. Heni, 2018; Toth & Vigo, 2014, Chapter 15). To the best of our knowledge, the first approach including alternative fueling vehicles with a limited driving range is the Green VRP (GVRP) proposed by Erdoğan & Miller-Hooks (2012). Although they do not focus on the use of EVs, they provide an initial framework for these types of problems. Davis & Figliozzi (2013) study the impact of replacing ICEVs by EVs for distribution and Felipe et al. (2014) extend the GVRP by introducing the partial-recharge policy, allowing vehicles to determine the battery charged at each fueling station, as well as multiple recharging technologies by assigning a different recharging speed to each station.

Recently, a stream of articles considers explicitly distribution problems using a fleet of EVs, where the energy from the battery of the vehicle is consumed as the vehicle operates, that may be replenished in order to eventually extend the length of the tour. The EVRP with time windows (EVRPTW) is proposed in Schneider et al. (2014), where the GVRP is extended with limited battery constraints and other classical VRP constraints (e.g. time windows and vehicle capacities), and tackled via metaheuristics. The discharge of the battery is assumed to be linear with respect to the distance traveled, a simplification of the real conditions. Desaulniers et al. (2016) develop a BCP algorithm for a generalization of the EVRPTW where full and partial recharge policies are allowed, as well as the possibility of limiting the number of en route recharges. Similarly, Roberti & Wen (2016) tackle the single-vehicle version of the problem. Partial recharges are also considered in Keskin & Çatay (2016), while Schiffer & Walther (2017) also incorporates location decisions. We remark that these problems can be formulated as a special case of VRP with intermediate stops, a more general family of VRPs recently surveyed in Schiffer et al. (2019).

Another interesting, complementary area of research focuses on enriching the EVRPTW and other variants by incorporating characteristics to reduce the gap with real-world operations. Demir et al. (2014) present a review of different energy consumption models. More complex and realistic energy consumption models for electric batteries, based on tests conducted using real data, are proposed by Goeke & Schneider (2015) and Wu et al. (2015). We remark that Goeke & Schneider (2015) extend the EVRPTW to consider a mixed fleet composed of both EVs and ICEVs. Their model incorporates the dependency on the travel speed, the mass of the vehicle, and the weight of the load to be transported into the energy consumption, among others. Further interesting results are reported by Fetene et al. (2017), where the effect of different factors on the energy consumption rate and the driving range is measured through a detailed analysis of a large-scale dataset of driving patterns collected from more than 700 EVs over a period of two years. The EVs correspond to private vehicles instead heavy-duty vehicles, and several factors are taken into account, such as travel speed, driver behavior, weather conditions, road type, and others, when considering their use. Regarding the travel speed, their results suggest that the energy consumption is non-linear and that the optimal driving speed has a *sweet spot* between 45 and 56 kilometer per hours, lower than the 65 kilometer per hours reported therein for the ICEVs. When comparing the characteristics of each model,

we notice differences in energy consumption, particularly at lower travel speeds. This suggests that additional research is required in this area to fine-tune the energy consumption models.

Recently, and simultaneously with our work, some approaches investigate algorithms for more complex battery discharge functions. Zang et al. (2022) consider a non-linear battery depreciation model to account for the impact of the *depth-of-discharge* and develop a column-generation-based heuristic with a tailored labeling algorithm. Rastani & Çatay (2023) tackle a research question that is similar to ours but focused on weight. They adjust the consumption model from Goeke & Schneider (2015) to account for the load, propose two compact ILP formulations, and propose an Adaptive Large Neighborhood Search (ALNS) heuristic to investigate the impact of the load in the energy consumption.

The interaction between the vehicles and the recharging infrastructure introduces new interesting operational constraints, with a direct impact on the quality and the feasibility of the solutions. These constraints are motivated by practical contexts and, in general, have been tackled independently. Sassi et al. (2014) consider time-dependent charging costs, multiple recharging technologies, and a mixed fleet of vehicles, modeling the dynamic pricing strategies in smart grid networks. Bessi et al. (2022) develop a BP algorithm considering multiple recharging technologies and propose a bi-directional labeling algorithm, still assuming he recharging time to be proportional to the amount of recharged energy. Montoya et al. (2017) focus on the charging process, observing that the amount of energy charged depends not only on time spent but also on the preliminary battery energy level. The charging functions modeling the process are nonlinear, and they show that continuous piecewise linear functions with up to 3 pieces translate into good approximations. This problem is also studied by Froger et al. (2019), introducing new algorithms that improve the quality of the solutions. Another interesting aspect deals with access to the recharging infrastructure. The limited capacity of a recharge station (in terms of chargers) may be exceeded by the high demand of vehicles during peak hours. This is considered in Keskin et al. (2019), where the vehicles cannot cannot assume to have access to a charger immediately upon arrival to a station, and therefore incur in waiting times that affect the rest of the route. Their experiments suggest that waiting times can result in up to a 26% difference in terms of costs. Similarly to the previous cases, the waiting times can be modeled by continuous piecewise linear functions that satisfy the First-In-First-Out (FIFO) condition. Overall, we refer the reader to Pelletier et al. (2016) for a complete survey of the literature considering modeling and algorithmic aspects.

Congestion effects on routing decisions have received increasing attention during the last few years, as they represent a critical aspect regarding last-mile deliveries in large cities (see e.g. Savelsbergh & Van Woensel, 2016). Time-Dependent VRPs (TDVRPs, see Gendreau et al., 2015) is the name assigned to a wide family of interesting optimization problems that incorporate explicitly the traffic conditions at a planning level by assuming that the travel time between two customers is variable and does not remain constant during the planning horizon. Regarding congestion, the model proposed by Ichoua et al. (2003) has been widely accepted as a standard within the VRP community. Briefly, variable travel speeds are modeled as a step function over the planning horizon, and the vehicle moves along the network according to these speeds depending on the departure time from a customer. As a result, the travel time between two customers is a continuous piecewise linear function that satisfies the FIFO condition. From an algorithmic perspective, TDVRPs require more complex models, algorithms, and implementations to handle variable travel times. We restrict our review to exact algorithms, although (meta)heuristic approaches have also been proposed recently in the related literature. Classical objective functions consider the makespan or, complementary, the duration of the route, incorporating in this fashion the departure time from the depot a decision variable. Single vehicle problems are usually referred to as the Time-Dependent Traveling Salesman Problem (TDTSP). The TDTSP minimizing the makespan is studied in Adamo et al. (2019); Cordeau et al. (2014), while algorithms for the variant with time windows and also minimizing the duration are studied in Lera-Romero & Miranda-Bront (2019); Lera-Romero et al. (2020a); Vu et al. (2020). Regarding Time-Dependent VRP (TDVRP), Dabia et al. (2013) develop a Branch-and-Price (BP) algorithm for the TDVRP with time windows (TDVRPTW) where the objective function minimizes the total duration of the routes. The pricing problem is tackled using tailored labeling algorithms that incorporate the time dependency. As a result, some of the resources are modeled via a function that depends on the time. The labeling algorithm applies a *total dominance criterion*, meaning that one label can dominate another only if this function is dominated over its entire domain. The TDVRPTW with Pickup and Deliveries (TDVRPTWPD) is studied in detail by Sun et al. (2018) using similar ideas, while a variant of the TDVRPTW with path flexibility is explored by Huang et al. (2017). Another problem having a similar structure is studied in Tagmouti et al. (2007), where an arc-routing problem with time-dependent piecewise linear service costs is transformed into a VRP, also considering a time-dependent service function. Regarding the solution approach, the labeling algorithm developed for the pricing problem resembles the one proposed in Dabia et al. (2013), which considers a total dominance criterion. Recently, Lera-Romero et al. (2020b) approach the TDVRPTW with a BCP algorithm that incorporates partial dominance within the labeling algorithm when solving the pricing problem, enabling a label to be dominated only at a portion of its domain. This enhancement shows to be very effective, outperforming previous approaches in the computational experiments.

The literature connecting the EVRPTW and the TDVRP is somehow scarce, and we highlight some recent research in this direction. Fukasawa et al. (2018) studies a complex optimization problem where, in addition to the routing decisions, the speed of the vehicles in each instant is also a decision variable in order to minimize the costs associated with fuel consumption. Pelletier et al. (2019) consider an optimization problem under the uncertainty of some external variables such as road friction, vehicle speed, and weather by estimating the deviation of the discharge rate in typical scenarios. Shao et al. (2017) enhance the classical EVRPTW by adding time-dependent travel times to obtain more accurate estimations regarding potential violations of the time windows, although the battery discharge model remains simplified and is not affected by the variable travel speeds. Recently, and simultaneously with our work, Lu et al. (2020) take the first steps towards a model that considers the impact of the time-dependent speeds on energy consumption. However, they assume that the congestion can be captured with a step function having only three pieces with different travel speeds. Besides the limitations regarding the applicability in practice, the proposed model does not scale for more general contexts. As far as we know, no article in the literature proposed a general model integrating the effects of congestion into the battery discharge model for EVs.

1.2. Our contributions

We build upon the research by Goeke & Schneider (2015) and Ichoua et al. (2003) to study the TDEVRPTW, a generalization of the EVRPTW that captures the effect of congestion on both the travel times and the battery consumption. Aligned with the observations raised by Desaulniers et al. (2016), considering explicitly the speed profiles as an input for the battery discharge model introduces an additional complexity regarding REFs compared to the

EVRPTW. Although the speed is not itself a decision variable, the routes must find implicitly a trade-off between traveling an arc at a less congested period to reduce the travel time, likely to be beneficial regarding the time windows, and the eventual increase in the energy consumption, arguably negatively affecting the driving range.

However, our conception of time-dependency goes beyond the classical time-dependent travel times, as we propose a unifying framework that naturally integrates further operational constraints such as the variable charging times (Montoya et al., 2017) and the waiting times (Keskin et al., 2019), both of which are approximated via continuous piecewise linear functions in their approaches. To the best of our knowledge, our paper is the first one to consider all these aspects in an integrated fashion from the routing perspective. Through our experimental setup, we provide valuable managerial insights on the impact of congestion in the driving range. For some of the scenarios considered, up to 40% of the routes obtained in a time-independent setup become infeasible due to the impact of congestion on the battery.

In terms of methodology, we design a BCP algorithm including state-of-the-art components. We develop a labeling algorithm capable of handling piecewise linear functions within each label to encode the speed-dependent energy consumption as a resource. Our labeling algorithm applies partial dominance rules to accelerate its execution. The BCP is enhanced with new, tailored preprocessing rules for the TDEVRPTW that can be further exploited by other VRP variants considering a limited driving range. Regarding the branching, we propose an alternative scheme for routing with EVs that simplifies the one proposed in Desaulniers et al. (2016) and, furthermore, reduces the number of enumerated nodes in a BP. Extensive computational experiments demonstrate that our approach is robust for the time-dependent instances, that our more general framework is efficient on time-independent EVRPTW benchmarks, and that it is able to solve time-dependent instances roughly of the same size as for the EVRPTW with comparable computational effort.

The rest of the paper is organized as follows. Section 2 defines the TDEVRPTW in its general fashion and introduces the notation used in the paper. Then, Section 3 presents a general framework encoding within the TDEVRPTW further time-related operational constraints. The BCP algorithm is presented in Section 4. Finally, the numerical experiments are presented in Section 5 with conclusions and a discussion of the possible future directions in Section 6.

2. Time-dependent electric vehicle routing problem

In this section we define the TDEVRPTW in its general fashion and provide a brief analysis of its novel characteristics. We review the variants from the literature that are incorporated later into the algorithms described in Section 4.2. Finally, we introduce the new time-dependent battery consumption model, which integrates the variable speeds into the battery discharge functions.

2.1. Problem definition

The TDEVRPTW is defined on a directed graph $D = (V, A)$ where each vertex $i \in V$ represents a location that can be either the depot, a customer, or a (recharging) station. Thus, $V = \{o, d\} \cup V_c \cup V_s$, where o and d are two vertices representing the depot, V_c is the set of customers, and V_s the set of recharging stations. Each arc $(i, j) \in A$ represents a path between locations i and j, where c_{ij} denotes its travel cost.

Classical routing problems incorporate real-life operational constraints such as vehicle capacity, time windows, and service times. We assume that an unlimited fleet of homogeneous EVs is available, where each vehicle has a capacity Q. Each customer $i \in V$ has a positive demand q_i, a time window $[a_i, b_i]$ where the service must start, and a service time s_i to process the customer.

Congestion effects are captured following the model proposed by Ichoua et al. (2003). Operations take place within a planning horizon $[0, T]$, which typically represents the length of a working day. Each arc $(i, j) \in A$ has an associated distance d_{ij} and a *speed function* $v_{ij}(t)$ indicating the speed of any vehicle traversing arc (i, j) at time t. In this model, $v_{ij}(t)$ is a step function of known, fixed average speeds defined over a partition of the planning horizon given as input. Then, a *travel time function* $\tau_{ij}(t)$ is derived for each arc $(i, j) \in A$, indicating the travel time of the trip from i to j if departing at time $t \in [0, T]$. Function $\tau_{ij}(t)\colon [0, T] \to \mathbb{R}$ is continuous, piecewise linear, and satisfies the FIFO property, meaning that delaying the departure cannot lead to an earlier arrival time.

Each EV is equipped with a battery with a limited capacity B (expressed in kWh). For each arc $(i, j) \in A$ we consider an *energy consumption function* $\beta_{ij}(t)$ that indicates the energy consumed while traveling from location i to j if departing from i at time t. As described in the next section, the consumption depends on the travel speed and, therefore, becomes a time-dependent function on the departure time from the origin location. Although the battery is discharged while the vehicle is in motion, it can be recharged at any station to continue the route. For an EV with an initial battery level of $w_0 > 0$ kilowatt hour and a recharge station $j \in V_s$, the charge obtained after t units of time is given by a function $g_j(w_0, t)$. Following the notation introduced by Desaulniers et al. (2016), the classical EVRPTW is retrieved by setting the battery-discharge function $\beta_{ij}(t) = b_{ij}$ and the recharging function $g_j(0, h_{ij}) = b_{ij}$ when assuming constant travel time functions, being b_{ij} the energy consumption for arc $(i, j) \in A$ and $h_{ij} = \alpha b_{ij}$ the time required to recharge b_{ij} units of energy. Section 2.2 provides a deeper analysis of these functions.

Most of the literature related to EVs allows recharges of either one of two policies: full or partial. Whereas the *full-recharge* policy forces all vehicles to charge batteries up to their full capacity at each station, the *partial-recharge* scenario provides a more granular choice. Partial recharge extends the solution space by indicating not only the path but also the amount of battery recharged at each station visited within the route. Desaulniers et al. (2016) also limits the number of en route recharges to one. This policy is called the *single-recharge* policy, while the former is referred to as the *multiple-recharge* policy.

A route $r = (0, v_1, \ldots, v_k, d)$ is a sequence of locations starting from the initial depot o and ending at the final depot d. Similarly to the EVRPTW, a route r is *feasible* if and only if: (i) there are no repeated customers (although a recharge station may be visited more than once); (ii) the capacity of the vehicle is not exceeded; (iii) customers are visited within their time window $[a_i, b_i]$; and (iv) the energy required $\beta_{ij}(t)$ for each arc (i, j) in a route must never exceed the battery level of the EV when departing from location i towards j. The last two conditions depend on the timing of the route, as both the travel time and the battery consumption depend on the speed that, in turn, depends on the time of departure from each of the traversed vertices. Then, as discussed in the next sections, the departure times from the depot and from the intermediate locations within a route become a decision variable. The cost of a route r is given by $c_r = \sum_{(i,j)\in r} c_{ij}$. The TDEVRPTW involves finding a set of feasible routes visiting each customer exactly once at the minimum total cost.

We further remark that while most time-dependent models have an objective function that minimizes the makespan or duration of the routes, we consider a more classical, time-independent objective function. This decision is motivated by several reasons, both regarding the model as well as its algorithmic implications.

Table 1
Variable definitions and suggested values as proposed by Demir et al. (2014). Vehicle mass and surface from the city delivery truck Alke ATX 320E.

Notation	Value	Description
g	9.81 meters per square second	Gravitational constant
ρ	1.42 kilograms per cubic meter	Air density
cr	0.006	Coefficient rolling resistance
cd	0.9	Coefficient aerodynamic drag
FS	2.40 square meters	Frontal surface
m_c	900 kilograms	Curb mass

We defer this discussion to Section 4.4.5, in order to provide better reflect the motivation behind this decision.

2.2. Battery consumption model

We consider the model proposed in Goeke & Schneider (2015), where the battery consumption depends on different variables grouped into three categories: vehicle mass, speed, and conditions of the terrain. Let $P(v, q)$ be the *instantaneous consumption function* of a vehicle traveling at speed v with load q, defined by the following equation

$$P(v,q) = \left(\frac{1}{2}\cdot cd\cdot\rho\cdot FS\cdot v^2 + (m_c + m_u q)\cdot g\cdot(\sin(\alpha) + cr\cdot\cos(\alpha))\right)\cdot v. \quad (1)$$

The rest of the parameters are defined in Table 1. Note that the mass m_u of each of the q units loaded in the vehicle is not specified. This variable depends on the type of goods to be delivered, which is problem-dependent. Similarly, the variable α indicates the gradient of the terrain, which depends on the road conditions.

Since we focus on the effects of the congestion, we adapt this formula to achieve a reasonable compromise between the simplicity and expressiveness of the model. We assume that the mass of each unit is negligible compared to the vehicle mass ($m_u = 0$), and that the gradient is zero ($\alpha = 0$). Thus, we define $h_1 = \frac{1}{2}\cdot cd\cdot\rho\cdot FS$ and $h_2 = m_c\cdot g\cdot cr$, and rewrite $P(v, q)$ as follows:

$$P(v) = h_1 v^3 + h_2 v. \quad (2)$$

Table 1 shows the estimated values for each parameter suggested by Demir et al. (2014). Replacing these values in the former expression results in $h_1 \approx 1.54$ and $h_2 \approx 52.97$. Thus, we can infer that, under these conditions, $h_2 \approx 35h_1$. This relation becomes relevant to create synthetic instances: given a single discharge rate h (for instance, as considered in the EVRPTW), an instantaneous consumption function can be derived by setting $h_1 = \frac{h}{35}$ and $h_2 = h - \frac{h}{35}$, assuming $v = 1$ for the EVRPTW. Establishing an analogous connection between the average travel speed and the speed functions will later enable us to evaluate the effects of congestion when compared to the EVRPTW.

Observe that the instantaneous consumption function $P(v)$ indicates the energy consumed in an instant of time by a vehicle traveling at speed v. However, the battery discharge function $\beta_{ij}(t)$ defined in Section 2.1 refers to the total energy consumed during the traversal of arc $(i, j) \in A$ if departing from i at time t. To compute the battery discharge function $\beta_{ij}(t)$ we must combine the time-dependent travel times with the instantaneous consumption functions $P(v)$. If a vehicle departs from vertex i to vertex j at time t, then it arrives j at time $t + \tau_{ij}(t)$. Thus, the vehicle is traversing arc (i, j) during interval $[t, t + \tau_{ij}(t)]$ and the battery discharge $\beta_{ij}(t)$ can be computed by integrating the instantaneous consumption function over that interval of time.

$$\beta_{ij}(t) = \int_t^{t+\tau_{ij}(t)} P(v_{ij}(x))dx \quad (3)$$

Recall that the speed function $v_{ij}(t)$ over arc $(i, j) \in A$ is a step function; let $|v_{ij}|$ be its number of pieces. If the kth piece of v_{ij} ($k = 1, \ldots, |v_{ij}|$) is defined on the domain $[T_k, T_{k+1})$ and has a constant value $\bar{v}_{ij}^k$, then Eq. (3) can be rewritten as

$$\beta_{ij}(t) = \sum_{k=1}^{|v_{ij}|} P(\bar{v}_{ij}^k) \times \left|[T_k, T_{k+1}) \cap [t, t + \tau_{ij}(t)]\right| \quad (4)$$

Similarly to the case of the travel time functions, the battery discharge functions are piecewise linear and continuous, as stated in the following proposition.

Remark 1. Given an arc $(i, j) \in A$, the discharge function $\beta_{ij}(t)$ is piecewise linear and continuous.

Proof. The result follows since (4) is a composition of continuous piecewise linear functions. □

To fully characterize the battery discharge functions, it is necessary to describe their breakpoints. We remark that τ_{ij} is computed in a similar fashion as β_{ij}, and therefore these breakpoints are the same for τ_{ij} and β_{ij} (Ichoua et al., 2003).

Fig. 1 presents an example of the different functions involved in the battery discharge model. Consider an arc $(i, j) \in A$ with a distance $d_{ij} = 2$ and a speed function v_{ij} as specified in Fig. 1(a). Furthermore, the fleet of EVs has batteries with an instantaneous consumption function $P(v) = \frac{1}{6}v^3 + \frac{1}{3}v$ (Fig. 1(c)). Then, the resulting travel time function τ_{ij} and battery discharge function β_{ij} are illustrated in Fig. 1(b) and (d), respectively.

Finally, recall the discussion from Section 1.1 regarding the differences between Goeke & Schneider (2015) and Fetene et al. (2017) in terms of the energy consumption and its dependence on travel speed. Although we concentrate on the model proposed by Goeke & Schneider (2015), we highlight that our approach is general enough to incorporate other discharge functions by defining the appropriate instantaneous consumption function of the average travel speeds.

2.3. Delayed departure from the locations

Frequently, tactical plans assume that the vehicles leave the customer immediately after the service is completed. This is the case in most of the classical variants of the TDVRPs, motivated by the fact that the travel times satisfy the FIFO property and, therefore, delaying the departure cannot improve the solution.

From the model presented in Section 2.2, the EVs require a different approach. The travel speed (and, therefore, the travel time) and the battery level are usually inversely correlated, meaning that a feasible route (eventually, the optimal one) may benefit from traveling at a slower speed, still satisfying all the operational constraints, in order to increase the driving range of the EV. To illustrate this situation, recall the example from Fig. 1. If the vehicle departs from i at $t_i = 7$, then the travel time is $\tau_{ij}(7) = 0.5$, and arrives at j at time 7.5 with a battery consumption $\beta_{ij}(7) = 6$ units of energy. However, if feasible, delaying the departure up to $t_i' = 8$ increases the travel time $\tau_{ij}(8) = 1$, arriving at j at time 9, but consuming less energy since $\beta_{ij}(8) = 2$. Note that this reduction in energy consumption may be critical regarding visits to other customers, and thus necessary for the optimal solution. Then, the model must allow a vehicle to delay its departure from a location in order to guarantee optimality, becoming also a decision variable and increasing the difficulty. In what follows, we provide an interpretation for this characteristic and discuss a potential practical implementation.

In order to avoid solutions including unrealistic waiting times, we limit the deferral of the departure to $b_i + s_i$ when visiting $i \in V$ (eventually, up to T for a station $i \in V_s$). We acknowledge that, even

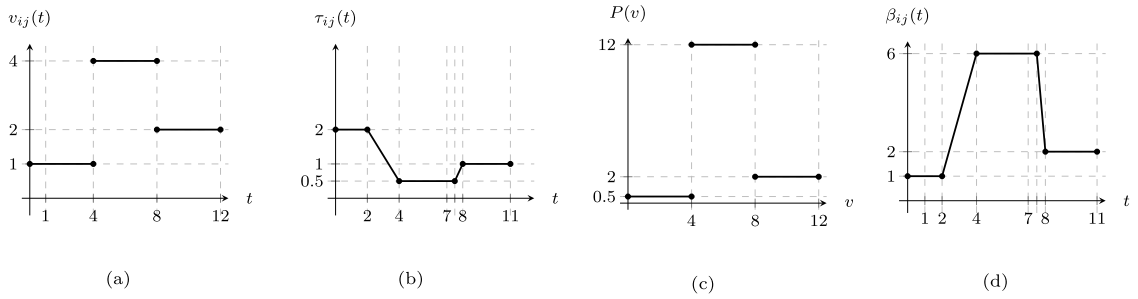

Fig. 1. Numerical example illustrating the speed function $v_{ij}(t)$ (1a); travel time function $\tau_{ij}(t)$ (1b); instantaneous consumption function $P(v)$ (1c); and battery discharge function $\beta_{ij}(t)$ (1d) for an arc $(i, j) \in A$ having $d_{ij} = 2$.

with this limitation, these adjustments may be difficult to implement in practice. However, there are some interesting interpretations from a managerial standpoint. First, they can be seen as a buffer regarding potential unexpected delays during the operations. Second, these waiting times suggest that the speed could be reduced when executing the route. To illustrate this second idea, recall the previous example and assume that the vehicle is ready to depart from i at $t_i = 7$, but the optimal solution indicates to delay the departure to $t'_i = 8$. Then, instead of delaying the departure, the vehicle may still leave at $t_i = 7$ but aim to arrive at j at time 9, as would be the case when departing at $t'_i = 8$. Simplifying the example, the average travel from i to j at average travel speed $\hat{v}_{ij} = 1$ in practice, without violating any time constraints. As mentioned before, we assume in this case that slower speeds consume less battery.

Note that this is not determined at the planning stage when solving the TDEVRPTW as the travel speeds and congestion patterns are not a decision variable, but rather an input for the problem. This decision is partially motivated by the fact that in practice, similarly to other VRPs, en route adjustments are usually applied in real-time due to deviations from the planned schedule. For the TDVRPs in general, and the TDEVRPTW in particular, this is particularly important as congestion is a dynamic phenomenon that may suffer from significant variations with respect to the standard patterns due to unexpected events, beyond the classical rush hour congestion. Then, travel speed adjustments originating from a delayed departure could be implemented during the execution rather than determined at the planning stage, in real-time, by suggesting to the driver a desired (eventually, optimal) range for the average travel speed for a given link. In this fashion, the suggested speed range may account not only for the updated, more detailed congestion estimation but also for the specific information about the current operational context, such as the timing and potential delays, among others.

From an algorithmic standpoint, suggesting an optimal travel speed range requires solving an additional, real-time optimization problem. Fukasawa et al. (2018) tackle a joint vehicle routing and speed optimization (JVRSP) for the VRP considering traditional ICEVs. If tailored for routing with EVs, this integrated problem could stand as the building block of a real-time decision support tool to complement the TDEVRPTW. First, we note that the JVRSP is a very challenging problem, both theoretically and practically. Second, the re-optimization of a route must incorporate all the operational constraints from the TDEVRPTW. This includes the battery discharge model, eventually with more detailed and updated travel speed information. Also, time windows must still be satisfied, imposing a lower limit for the speed reduction implicitly. If the energy consumption model is aligned with the pattern observed in Fetene et al. (2017), suggestions should further either implicitly or explicitly incorporate a minimum recommended travel speed, as traveling too low may not be beneficial for the time windows nor for the driving range. Note that the re-optimization may also require increasing the travel speed if the current execution is behind schedule in order to meet the time windows, eventually introducing additional recharges, if necessary. Finally, note that this framework can be further generalized to deal with other types of events, eventually becoming a real-time disruption management tool.

2.4. Battery consumption and feasible routes.

Having defined the battery consumption model at an edge level, we next proceed to extend these definitions to routes. This represents a key concept within our model, as the feasibility of a route depends on maintaining a positive battery level during the entire trip. Let $p = (o = v_1, \dots, v_k)$ be a path starting at the depot and t be a potential *ready time* at vertex v_k, that is, a time instant at which the vehicle has visited (processed) all the vertices in p and is ready to depart to the next vertex (or, alternatively, finish the trip if $v_k = d$). Both t and the battery level at t depend not only on the sequence of vertices in p and the departure time from the depot, but also on the other interdependent decisions taken along the traversal of p. Different battery levels could be feasible for a given time instant t depending on the energy consumption while traversing p, which in turn depends on the travel speeds and the departure times; the amount of energy recharged en route; and eventually on the delayed departure discussed in the previous section.

Note that the battery level, including both the consumption and the recharges, does not affect the cost of a route (and, therefore, the overall solution) as it only restricts its feasibility. Then, given a path p and a time t, we focus on the combinations of decisions that lead to the maximum battery level when p is ready at time t, discarding other dominated solutions. In what follows, we develop the model considering a partial-recharge policy, although it can be easily adapted to other contexts. Similarly to the classical time-independent variants, we assume that the service time s_i for vertex i is already encoded into the travel time function $\tau_{ij}(t)$. Specific details are provided in Section 3. We first consider a simplified definition for a given arc or vertex, and then we generalize the idea for paths.

Suppose we are given a continuous and piecewise linear function λ such that $\lambda(t)$ indicates the *maximum battery level* of a vehicle that is ready to depart a vertex i at a time t. Moreover, suppose the domain of λ is a closed interval $\mathrm{dom}(\lambda)$. When traveling through an arc $(i, j) \in A$ and departing from i at time t, the battery consumed is given by $\beta_{ij}(t)$ and, therefore, the battery level when arriving at j is by $\lambda(t) - \beta_{ij}(t)$. However, this need not be the maximum battery level achievable to visit j at time $t + \tau_{ij}(t)$ when traveling directly from i. Indeed, the vehicle may have arrived at j at a time t' earlier than t, with more charge in the battery, and waited until time t (which does not affect the battery level). We define the function $TRV^{\lambda}_{ij}(t)$ to capture this behavior. That is, $TRV^{\lambda}_{ij}(t)$ denotes the *maximum battery level* when arriving

at j at a time t after traversing the arc $(i, j) \in A$, given that the maximum battery level function at i is λ. For each time t, let $H(t)$ be the set of times t' such that j is reached at time t or earlier when i is departed from at time t'. That is, $H(t) = \{t' \in \text{dom}(\lambda) \mid \max\{a_j, t' + \tau_{ij}(t')\} \leq t\}$. Then, $TRV_{ij}^{\lambda}(t)$ is defined by

$$TRV_{ij}^{\lambda}(t) = \max_{t' \in H(t)} \left\{\lambda(t') - \beta_{ij}(t')\right\}. \tag{5}$$

Recall that functions λ, β_{ij} and τ_{ij} are continuous and piecewise linear, and τ_{ij} also satisfies the FIFO property. Thus, $\lambda - \beta_{ij}$ and $t \to \max\{a_j, t + \tau_{ij}(t)\}$ are also continuous and piecewise linear. By definition, $H(t') \subseteq H(t)$ for $t' \leq t$, thus $H(t)$ is a closed interval $[\underline{h}(t), \bar{h}(t)]$ because τ_{ij} satisfies the FIFO property. Note that $\underline{h}(t)$ is a constant function, whereas $\bar{h}(t)$ is continuous and piecewise linear, hence the domain $\text{dom}(TRV_{ij}^{\lambda})$ of TRV_{ij}^{λ} is a closed interval $[t_1, t_2]$ as well. For the sake of simplicity, we abuse notation and define a restricted domain considering arrival times at j such that the maximum battery level is non-negative, i.e. $\text{dom}(TRV_{ij}^{\lambda}) = \{t \in [t_1, t_2] : TRV_{ij}^{\lambda}(t) \geq 0\}$, that still remains a closed interval. This definition removes the infeasible arrival times at j that cannot be reached because the battery is depleted. Note that $TRV_{ij}^{\lambda}(t)$ implicitly encodes waiting times incurred by eventually delaying the departure from j, as it considers feasible ready times earlier than t.

A similar analysis holds for the stops at the recharging stations. In this case, we further assume the recharging functions $g_i(w_0, t)$ to be continuous and piecewise linear and provide in Section 3 the motivation for this hypothesis. Once again, suppose we are given a continuous and piecewise linear function μ such that $\mu(t)$ indicates the maximum battery level of a vehicle that is ready to start a recharge at time t in a station $i \in V_s$. Again, μ is defined over a closed interval $\text{dom}(\mu)$ and assume that the image of μ is included in $\text{dom}(g_i)$. Let $CHG_i^{\mu}(t)$ be the *maximum battery level* when the recharge is completed at time t, defined as

$$CHG_i^{\mu}(t) = \max_{t' \leq t, t' \in \text{dom}(\mu)} \left\{\mu(t') + g_i(\mu(t'), t - t')\right\}. \tag{6}$$

Again, as different battery levels are feasible for a given t, $CHG_i^{\mu}(t)$ indicates the maximum. Since recharging has no cost, it is always convenient to charge instead of waiting (without recharging). Similarly to the previous case, the function $CHG_i^{\mu}(t)$ also captures possible delays incurred by waiting times, while the domain $\text{dom}(CHG_i^{\mu})$ of CHG_i^{μ} is equal to the closed interval $\text{dom}(\mu)$.

Both TRV_{ij}^{λ} and CHG_i^{μ} are defined in terms of piecewise linear functions, although their structure is slightly more complex than a straightforward composition. Based on the previous approach, the function G in the following result generalizes the structure of TRV_{ij}^{λ} and CHG_i^{μ}. The detailed proof can be retrieved from Appendix A.

Proposition 1. *Let f be a continuous and piecewise linear function, and b be a continuous and non-decreasing function. Define a function G on a domain $[x, y]$ as*

$$G(t) = \max_{t' \in [b(x), b(t)]} f(t', t), \tag{7}$$

where $[b(x), b(y)] \subseteq \text{dom}(f)$. Then, G is continuous, piecewise linear, and non-decreasing.

Proposition 1 is relevant from a practical standpoint, as it enables to compute TRV_{ij}^{λ} and CHG_i^{μ} considering only a finite subset of breakpoints. We summarize this in the following results; note that b is piecewise linear in the definitions of TRV and CHG.

Corollary 1. *Let $|G|$ and $|f|$ be the number of pieces of G and f in $[x, y]$ and $[b(x), b(y)]$, respectively. Then, $|G| \leq |f|$.*

Corollary 2. *If b is piecewise linear, then G can be computed in polynomial time. If the pieces of f and b are sorted, then G can be computed in linear time.*

We now extend these ideas to paths, in order to compute the battery level at any given moment of a trip. Let $p = (o = v_1, v_2, \ldots, v_k)$ be a path starting at the depot o, but not necessarily ending at d. Define $\lambda_{v_i}^{p}(t)$ as the maximum battery level at a ready time t when visiting vertex v_i along path p, $i = 1, \ldots, k$. Recall that t is a ready time when v_i can be departed from at time t, meaning that v_i has already been "processed". Certainly, each $t \in [0, T]$ is a ready time for the depot v_1. Thus, as each vehicle departs from v_1 with a full battery level, it follows that $\lambda_{v_1}^{p}(t) = B$ for every $t \in [0, T]$. Clearly, $\lambda_{v_1}^{p}$ is a continuous and piecewise linear function defined over a closed interval. To define $\lambda_{v_i}^{p}$ we proceed by induction on i, taking advantage of the functions TRV and CHG. Let $\lambda = \lambda_{v_{i-1}}^{p}$, i.e., $\lambda(t)$ is the maximum battery level when vertex v_{i-1} is ready at time t. If v_i is customer, then each arrival time is a ready time, thus $\lambda_{v_i}^{p}(t) = TRV_{v_{i-1}v_i}^{\lambda}(t)$. Recall that $\lambda_{v_i}^{p}$ is a continuous and piecewise linear function defined on a closed interval, as it is required by the induction. If v_i is a recharging station, then v_i can be departed from only after the battery is recharged. Thus, we have to consider the time required to travel from v_{i-1} to v_i plus the time to recharge at v_i. Consequently, $\lambda_{v_i}^{p}(t) = CHG_{v_i}^{\mu}(t)$ where μ is the function such that $\mu(t) = TRV_{v_{i-1}v_i}^{\lambda}(t)$. Once again, $\lambda_{v_i}^{p}(t)$ is a continuous and piecewise linear function defined on a closed interval.

The functions $\lambda_{v_i}^{p}$ are used to encode the battery level within our labeling algorithm, as discussed in Section 4.4. We further provide a numerical example illustrating these definitions in Appendix B.

3. A general framework for time-dependent times

We next describe how to naturally integrate to the TDEVRPTW further time-related management aspects, previously studied in the literature in an isolated fashion, without changing the structure of the problem. Indeed, all these additional features can be directly encoded as a TDEVRPTW instance as described in the previous section.

3.1. Handling service times

Similar to time-independent problems, the service time s_i of a customer $i \in V_c$ can be directly encoded into the travel time and battery discharge functions to simplify the model and definitions. The following equations describe how to transform an instance $\mathcal{I}$ of the TDEVRPTW into an equivalent instance $\hat{\mathcal{I}}$ where the service times can be assumed to be zero. Let τ_{ij} and β_{ij} be the travel time and battery discharge functions for arc (i, j) in $\mathcal{I}$, and $\hat{\tau}_{ij}$ and $\hat{\beta}_{ij}$ the corresponding functions in $\hat{\mathcal{I}}$. Then, for $\hat{\mathcal{I}}$ set

$$\hat{\tau}_{ij}(t) := s_i + \tau_{ij}(t + s_i) \tag{8}$$

$$\hat{\beta}_{ij}(t) := \beta_{ij}(t + s_i). \tag{9}$$

and $s_i = 0$ for $i \in V$. This transformation assumes that the final depot s_d has no service time ($s_d = 0$). Moreover, note that the travel time functions still obey the FIFO property. To avoid confusion with the previous formulae, in the remaining of the document we assume that the service time is already encoded within the functions, but we maintain the original notation of τ_{ij} for the travel times and β_{ij} for the battery discharge.

3.2. Nonlinear charging times

Different approaches have been considered in the related literature for the battery recharging model. The EVRPTW, in its original setup, considers that the recharging time is linear to the amount

of energy to recharge, independently of whether the battery is empty or, for instance, half-full charged. According to Montoya et al. (2017), this recharging time is not linear. Moreover, they estimate that recharging the initial 75% of the battery requires a similar time than the final 25%. For each station, $j \in V_s$, let $\hat{g}_j(t)$ indicate the energy (in kWh) recharged in t units of time in a vehicle with an empty battery. They show that considering $\hat{g}_j(t)$ as a continuous concave piecewise linear function with three pieces results in a good estimation of the recharge times based on collected data. Moreover, this function is invertible.

Using these estimations, note that $g_j(w_0, t) = \min\{B - w_0, \hat{g}_j(\hat{g}_j^{-1}(w_0) + t) - w_0\}$ indicates the charge obtained after t units of time for an EV with battery level w_0. By definition, $g_j(w_0, t)$ is continuous and piecewise linear as required previously.

3.3. Time-dependent waiting times

As suggested by Keskin et al. (2019), the assumption that a vehicle can perform a recharge upon arrival to a charging station can sometimes lead to unrealistic solutions. This observation is relevant in the context of mid-haul and city logistics, as congestion is likely to occur at the recharging stations due to increased demand during peak hours. To capture this behavior, they introduce a time-dependent waiting time function $\omega_s(t)$ for each station $s \in V_s$ indicating the expected waiting time until a terminal is available if a vehicle arrives at station s at time t. These functions are continuous, piecewise linear, and satisfy the FIFO property.

Time-dependent travel times naturally incorporate these waiting times following a similar procedure to the one described for service times. For the sake of simplicity, assume $\omega_i(t) = 0$ for all $t \in [0, T]$ when $i \in V \setminus V_s$. Similarly to Section 3.1, given a vertex $j \in V$, then the travel time function for arc $(i, j) \in A$ is redefined as

$$\hat{\tau}_{ij}(t) := \tau_{ij}(t) + \omega_j(t + \tau_{ij}(t)). \tag{10}$$

Observe that this transformation assumes there is no distinction between travel and waiting times regarding the objective function. Otherwise, the waiting time functions may require to be handled explicitly. Again, to avoid confusion, we assume that the time-dependent waiting times are directly encoded in function τ_{ij}.

A key property behind the time-dependent travel time model lies in the FIFO condition. The following proposition states that the property holds after applying this transformation. The proof is omitted as it is the direct consequence of adding and composing continuous piecewise linear functions that satisfy the FIFO condition.

Proposition 2. *The FIFO property holds for the travel time functions after applying (10).*

However, a negative result is that the breakpoints defining τ_{ij} and β_{ij} may not remain the same after applying this preprocessing, as stated in Section 2.2.

We provide a numerical example to illustrate this transformation in Fig. 2. Let $j \in V_s$ be a station having waiting time function ω_j (Fig. 2(a)). Consider an arc $(i, j) \in A$, and let τ_{ij} be its travel time function (Fig. 2(b)). The preprocessed function τ_{ij} is depicted in Fig. 2(c).

Finally, we remark that a similar analysis can be developed to model time-dependent service times via continuous piecewise linear functions. Although we do not consider explicitly this scenario, they can be incorporated in a similar fashion to our framework. This may become relevant in practice, for instance, to model variable unloading times due to difficulties in finding parking spots during peak hours or similar use cases arising frequently in logistics.

3.4. Further infrastructure characteristics

Finally, note that some EVs admit multiple charging modes, such as different charging speeds or technologies for a given range of battery capacity. This can be modeled by defining a specific charging function for each case. Briefly, each station is associated with a specific *charging mode*, which indicates the available recharging function. For recharging stations having multiple modes, we can simply replicate the station. A fixed cost for each mode can be further incorporated into the inbound arcs to each station as well.

4. Exact algorithm

This section describes the main components of the exact BCP algorithm. We explain how to adapt the framework to toggle the different recharging policies and include a discussion regarding alternative objective functions. We highlight that our algorithm extends and generalizes the one proposed by Desaulniers et al. (2016) and constitutes a unifying framework for several interesting problems related to planning with EVs.

4.1. Preprocessing

4.1.1. Reducing arcs and time windows

Two interrelated steps are applied that search for infeasible arcs and shrink time windows, respectively. These steps are repeated while there are changes in the instance structure. For this procedure, we follow the ideas of Lera-Romero & Miranda-Bront (2019) that extend the classical rules proposed by Desrosiers et al. (1995). Briefly, an arc $(i, j) \in A$ is removed if is either capacity infeasible, i.e. $q_i + q_j > Q$, or time infeasible, i.e. $a_i + \tau_{ij}(a_i) > b_j$. In addition, we consider as input for the TDEVRPTW the preprocessed TDVRPTW instances, with the time windows already adjusted according to the time-dependent travel times. We omit the details for the sake of brevity but refer to Desrosiers et al. (1995) for a detailed description.

4.1.2. Minimum battery required

We next introduce a preprocessing rule specifically designed for the TDEVRPTW. For each vertex $i \in V$ we can compute a lower bound $MBR(i)$ of the *minimum battery required* to reach any recharge station (or eventually the depot) from vertex i without depleting the battery. Let $\underline{\beta}_{ij} = \min\{\beta_{ij}(t) \mid t \in \mathrm{dom}(\beta_{ij})\}$ be a lower bound on the energy consumption for arc $(i, j) \in A$. Then, for $i \in V$, a lower bound $MBR(i)$ can be computed via a standard shortest-path algorithm using $\underline{\beta}_{ij}$ as arc weights. Note that $MBR(i) = 0$ for $i \in V \setminus V_c$. In addition, arcs $(i, j) \in A$ satisfying $B - \underline{\beta}_{ij} < MBR(j)$ indicate that the battery level when reaching j cannot be enough to either reach another station or the depot, and therefore can be safely discarded. These bounds are used further to enhance the feasibility rules of the labeling algorithm to reduce the size of the enumeration tree, as described in Section 4.4.

4.2. Set-partitioning formulation

BCP algorithms and extended formulations based on the set-partitioning model stand as one of the most effective approaches to tackle different variants of the VRP. Let Ω be the set of all the feasible routes for the TDEVRPTW. For each route $r \in \Omega$, c_r represents its cost and the constant a_{ir} indicates if route r visits customer $i \in V$. Let y_r be a binary variable indicating whether a route $r \in \Omega$ is selected in the optimal solution. The set-partitioning formulation for the TDEVRPTW is defined as follows.

$$\min \sum_{r \in \Omega} c_r y_r \tag{11}$$

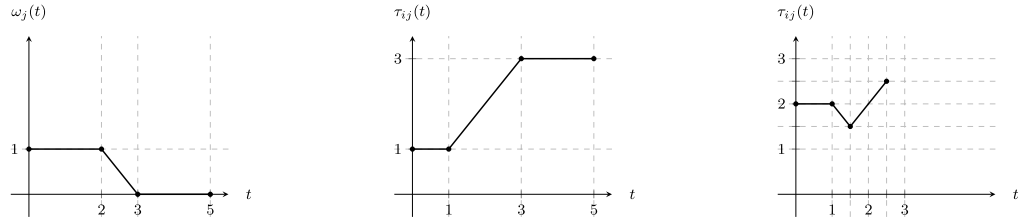

Fig. 2. Example for the preprocessed time-dependent waiting times including waiting time function $\omega_j(t)$ of $j \in V_s$ (Fig. 2(a)); $\tau_{ij}(t)$ of arc $(i, j) \in A$ (Fig. 2(b)); $\tau_{ij}(t)$ of arc (i, j) after (10) (Fig. 2(c)).

$$\text{s.t.} \sum_{r \in \Omega} a_{ir} y_r = 1, i \in V_c \tag{12}$$

$$y_r \in \{0, 1\}, r \in \Omega. \tag{13}$$

The objective function (11) minimizes the total costs of the solution, whereas restrictions (12) enforce that each customer is visited exactly once and (13) impose the domain of the variables.

Since Ω is exponential, the classical approach is to solve the LP relaxation at each node by using column generation. A restricted master problem (RMP) is initialized with a subset of the routes $\Omega' \subseteq \Omega$. For the TDEVRPTW, the initialization of the RMP is slightly more complicated compared to other variants as the traditional strategy selecting the routes $r = \{o, i, d\}$ to define Ω' may be infeasible due to the battery consumption. To overcome this issue, we initially include in Ω' an artificial (infeasible) route r that visits all customers with cost $c_r = n \times \max_{(i,j) \in A} c_{ij}$ to guarantee the feasibility of the LP relaxation of the RMP.

After the initialization step, new columns are added iteratively until the algorithm converges to the optimal (fractional) solution. At each iteration, the LP relaxation of the RMP is computed to obtain the dual variables π_i associated with constraints (12) for $i \in V_c$. Using this information, if a feasible route r with negative reduced cost $\bar{c}_r = c_r - \sum_{i \in r} \pi_i$ exists, then it is added to the RMP and the procedure is repeated. Otherwise, the current fractional solution is optimal. For the TDEVRPTW, the pricing problem becomes a Time-dependent Electric Elementary Shortest Path Problem with Resource Constraints (TDEESPPRC). The most effective exact approaches to tackle similar problems rely on dynamic programming and labeling algorithms (Desaulniers et al., 2016). Heuristics are in general used to speedup the performance of the pricing step, and exact algorithms are executed when no negative reduced cost routes are found to guarantee optimality (see Section 4.4.4).

4.3. Branching scheme

In this section, we review the standard ideas considered for the EVRPTW and propose a tailored branching rule to manage the intermediate stops at the recharging stations. Classical BCP algorithms consider branching on the so-called *arc variables*. Given an optimal solution y^* of the LP relaxation for (11)–(13), define for each arc $(i, j) \in A$ a binary variable $x^*_{ij} = \sum_{r \in \Omega, (i,j) \in r} y^*_r$. When no intermediate stops between customers are possible, such as in the VRP, it can be easily shown that y^* is fractional iff x^* is fractional as well. Based on this result, a *robust* branching rule can be implemented (see, e.g. Costa et al., 2019). We refer to this rule as the Arc Branching Rule (ABR).

Formulation (11)–(13) encodes the feasibility of a route, including the visits to the recharging stations, as part of the definition of Ω. Note that constraints (12) are enforced only for customer vertices $i \in V_c$, and indeed Ω can be restricted to consider at most one minimum-cost route for each set of customer vertices $S \subseteq V_c$, since all other routes visiting the same customers become dominated. From now on, we assume Ω satisfies this property as it can be easily implemented and is a key hypothesis for our branching.

Based on this observation, we propose an alternative branching rule exploiting the sequence of customers visited by a route, which we name Customer Branching Rule (CBR). Let $r = (0 = v_1, \ldots, v_k = d) \in \Omega$ be a feasible route. We define the *customer sequence of* r, $CR(r)$, as the sequence of customer vertices in r, removing those vertices $v_i \in V_s$. Given $i, j \in V_c$ and $r \in \Omega$, let b_{ijr} be a constant indicating whether j is visited immediately after i in $CR(r)$. Intuitively, $b_{ijr} = 1$ if there are no customers between i and j in route r. Let further define binary variables z_{ij} indicating if vertex j is visited immediately after i in any route $r \in \Omega$ disregarding recharge stations, that can be retrieved from the primal solution y as follows

$$z_{ij} = \sum_{r \in \Omega} b_{ijr} y_r. \tag{14}$$

Given a fractional variable z_{ij} we create two descending branches, one for $z_{ij} = 0$ and the other one for $z_{ij} = 1$. The following proposition proves that CBR is a valid branching rule (see Appendix C for a proof).

Proposition 3. *Let $(\bar{y}, \bar{z})$ be a solution of formulation (11)–(13) extended with variables z from (14). Then, $\bar{y}$ is a feasible solution for the TDEVRPTW if and only if $\bar{z}_{ij} \in \{0, 1\}$ for all $i, j \in V_c$.*

Simultaneously with our work, Bessi et al. (2022) briefly discuss the use of the CBR. We further provide specific details on our implementation of the CBR into our framework. Conceptually, the CBR can be handled in a similar fashion as the ABR. Constraint $z_{ij} = 1$ can be replaced by the union of constraints $z_{ik} = 0$ for $k \in V_c \setminus \{j\}$, and $z_{kj} = 0$ for $k \in V_c \setminus \{i\}$. This is particularly helpful when adapting the pricing problem to handle this branching strategy as explained in Section 4.4.2. Finally, Fig. 3(b) depicts the variables z_{ij} associated to the solution presented for Fig. 3(a).

4.3.1. Other branching alternatives

Generally, the ABR is combined with some additional rules to further reduce the enumeration tree. For the EVRPTW, Desaulniers et al. (2016) propose a four-level branching strategy. Firstly, they branch on the number of vehicles $\sum_{r \in \Omega} y_r$. Then, branching is performed on the total number of recharges $\sum_{r \in \Omega} \rho_r y_r$, where ρ_r indicates the number of recharges performed in the traversal of route r. The next decision is based on the number of recharges at a given station $\sum_{r \in \Omega} \rho_{rj} y_r$, where ρ_{rj} indicates the number of recharges of route r at station $j \in V_f$. Finally, ABR is considered by variables x_{ij} if none of the above are fractional. We remark that these rules can be combined with the CBR as well.

Fig. 3. Example for arc variables x_{ij} (ABR, 3a) and customer variables (CBR, 3b). The instance has depots 0 and 9, recharging stations $\{1, 2, 3\}$ and customers $\{4, \ldots, 8\}$ where a feasible fractional solution composed of four routes $r_1 = (0, 5, 3, 6, 1, 7, 8, 9)$, $r_2 = (0, 5, 3, 2, 4, 2, 9)$, $r_3 = (0, 3, 6, 1, 2, 4, 2, 9)$, $r_4 = (0, 2, 4, 2, 1, 7, 8, 9)$, with variable values $y_1 = \frac{2}{3}$ and $y_2 = y_3 = y_4 = \frac{1}{3}$ is shown. In the ABR, all arcs incident to a customer have integer values yet the solution is fractional.

4.4. The pricing problem

This section describes the labeling algorithm developed to tackle the pricing problem for Section 4.2 and the variants from Section 3. We also provide a brief discussion regarding the impact of considering alternative objective functions.

4.4.1. The labeling algorithm

We develop a forward labeling algorithm that explores all feasible paths. These algorithms generate implicitly an enumeration tree where each node is called a label and represents a partial path starting from the initial depot o. Aiming to overcome the exponential growth, pruning rules are in general incorporated to reduce the number of labels enumerated. In a time-independent context, each label is usually either discarded or processed. However, time-dependent problems require further consideration since labels encode some of their resource values as functions over a given domain. Thus, partial pruning rules eventually discard some specific intervals in the domain (see, e.g. Lera-Romero et al., 2020b).

Next, we present the forward labeling algorithm for the TDEVRPTW with a partial-recharge policy and unlimited recharges. Each path $p = (o, \ldots, v)$ is represented by a label $L = (prev, v, S, q, c, \lambda_v^p)$ where $prev$ is a pointer to the parent label, v is the last vertex, S is a set of unreachable customers, q is the total demand, c is the cost, and λ_v^p is the maximum battery level function defined in Section 2.4, i.e., $\lambda_v^p(t)$ is the maximum battery level at the ready time t after traversing p. For simplicity, let $prev(L)$, $v(L)$, $S(L)$, $q(L)$, $c(L)$, and λ_L denote the components $prev$, v, S, q, c, and λ_p^v of label L, respectively. In addition, let $p(L)$ be the path represented by L and let $\mathrm{dom}(L) = \mathrm{dom}(\lambda_L) = [a(L), b(L)]$ denote the feasible ready times at $v(L)$ after traversing p, with $a(L) = \min(\mathrm{dom}(L))$ being the earliest arrival time to vertex $v(L)$. When $p(L)$ is a route we call L a *route label*.

Let $\tilde{c}_{ij} = c_{ij} - \pi_i$ be the adjusted arc costs including the dual variables from the RMP. Since vehicles depart from the depot with a full battery, the initial label is represented by tuple $(\perp, o, \{o\}, 0, 0, 0, \lambda_o)$, where $\lambda_o(t) = B$ for $t \in [a_o, b_o]$. The following rule indicates how to extend a label during the enumeration process.

Rule 1 (Extension)**.** Given a label L, the extension through an arc $(v(L), w)$ results in a new label L_w where $prev(L_w) = L$, $v(L_w) = w$, $S(L_w) = S(L) \cup \{w\}$, $q(L_w) = q(L) + q_w$, $c(L_w) = c(L) + \tilde{c}_{vw}$, and function λ_{L_w} is computed from λ_L as discussed in Section 2.4, i.e.

$$\lambda_{L_w} = \begin{cases} \lambda_w^{p(L_w)}(t) = TRV^{\lambda}_{v(L)w}(t) & \text{if } w \in V_c, \\ \lambda_w^{p(L_w)}(t) = CHG^{\mu}_w(t), \text{ with } \mu(t) = TRV^{\lambda}_{v(L)w}(t) & \text{if } w \in V_s. \end{cases}$$

Note that L_w is feasible when $w \notin S(L)$, $q(L_w) \le Q$ and $a(L) + \tau_{vw}(a(L)) \le b_w$.

Battery requirements enable to the introduction of additional tailored feasibility rules. Recall from Section 4.1.2 that given a vertex $v \in V$, $MBR(v)$ indicates the minimum battery required to depart from a vertex v and reach a station or, alternatively, the end depot without depleting the battery. The following rule exploits this information to discard some intervals from $\mathrm{dom}(L)$.

Rule 2 (Feasibility)**.** Given a label L, any time $t \in \mathrm{dom}(L)$ such that $\lambda_L(t) < MBR(v(L))$ can be safely discarded from λ_L. If $\mathrm{dom}(\lambda_L) = \emptyset$ then $p(L)$ is *infeasible* and L can be discarded.

Since function λ_L is non-decreasing by definition, any interval discarded from $\mathrm{dom}(\lambda_L)$ must be a prefix. This fact can reduce the time required to compute this rule for many labels.

Another action to reduce the number of labels enumerated relies in applying dominance rules. Briefly, given two labels L, M, if every feasible extension p' of $p(L)$ is also a feasible extension of $p(M)$ but $p(M) + p'$ always has a lower cost, then L can be discarded. In practice, to reduce the computational burden, sufficient conditions are tested to ensure that a label can be discarded. In a time-dependent context, partial dominance enables to discard some specific dominated time instants from the domain $\mathrm{dom}(L)$, even if the label is not entirely dominated. As a result, the domain of a label may become a collection of intervals instead of a single one. However, this can be mitigated by assuming the vehicle simply waits without consuming battery in this dominated interval, which is modeled by considering a constant function extending the function from the previous non-dominated piece (we refer to Lera-Romero et al., 2020b for details). To avoid introducing further definitions, we abuse notation and assume the battery level functions λ_L and λ_M, as well as their corresponding domains, are adjusted in this fashion in the following rule.

Rule 3 (Partial dominance)**.** Let L and M be two labels satisfying (i) $v(L) = v(M)$, (ii) $q(M) \le q(L)$, (iii) $c(M) \le c(L)$, and (iv) $S(M) \subseteq S(L)$. Then, any time $t \in \mathrm{dom}(\lambda_L) \cap \mathrm{dom}(\lambda_M)$ such that (v) $\lambda_L(t) \le \lambda_M(t)$ can be discarded from $\mathrm{dom}(L)$. If $\mathrm{dom}(\lambda_L) = \emptyset$ then we say L is *fully dominated* and can be safely discarded.

From an implementation standpoint, to accelerate the dominance tests we consider the classical bucket structure where a dynamic programming table stores labels indexed by their resources v, q. Within each bucket, labels are sorted by cost. Observe that dominance tests include a comparison of the battery level functions between two labels, which requires linear time with respect to the total number of pieces in both functions. For the sake of brevity, we skip the implementation details. Fig. 4 shows an example of the partial dominance algorithm when label L is being verified for dominance against label M, where the dashed line represents the constant extension over a dominated interval in the domain of label M. Fig. 4(b) shows the remaining pieces from L after the dominance.

4.4.2. Incorporating CBR to the pricing algorithm

The CBR branching rule imposed over variables z_{ij} can be managed within the pricing algorithm in a similar fashion as the

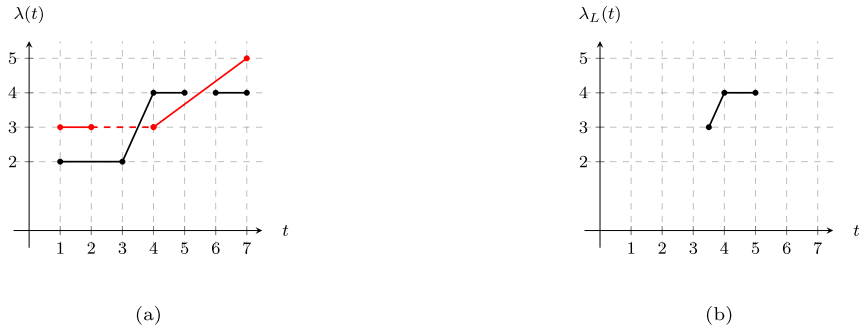

Fig. 4. Example of the partial dominance applied to L (black) and M (red) (4a) and the remaining pieces (4b). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

branching over the classical arc variables x_{ij} by preventing such routes from being generated instead of adding explicitly the constraints $z_{ij} = 0$.

At each node of the branch and bound tree, define $F_i = \{j \in V_c \cup \{d\} \mid z_{ij} = 0\}$ as the set of *forbidden successors* for each customer $i \in V_c \cup \{o, d\}$ (and let $F_i = \emptyset$ for each $i \in V_s$). Intuitively, F_i encodes the set of successors of i that cannot be visited due to branching decisions taken earlier in the tree. We include in the representation of a label L an additional resource u indicating the last vertex from $V_c \cup \{o, d\}$ visited in $p(L)$, i.e. either a customer or the depot. The extension of L through an arc $(v(L), w) \in A$ must additionally verify that $w \notin F_{u(L)}$. Consequently, Rule 3 must be modified accordingly to remain valid, where $F_{u(M)} \subseteq F_{u(L)}$ must also hold in order to conclude that label M dominates label L at time t. This new condition ensures that every feasible extension of L does not involve $w \in F_{u(M)}$ as the next vertex, and therefore is also a feasible extension of M. We further strengthen this condition by considering the set of already visited customers $S(L)$ as follows

$$F_{u(M)} \subseteq F_{u(L)} \cup S(L). \tag{15}$$

Incorporating condition (15) to Rule 3 affects the efficiency of the labeling algorithm. Similarly to the ABR for VRPs without intermediate stops between customers, label extensions are discarded because they may become infeasible. On the contrary, dominance rules become weaker due to the introduction of a new condition. Clearly, this implies that more labels are enumerated, although we remark that the effect of condition (15) is limited to only those labels L that end in a recharging station, otherwise $v(L) = v(M) = u(L) = u(M)$ and the inclusion is trivially satisfied.

4.4.3. Alternative recharge policies

So far, the battery consumption model and the labeling algorithm have been presented under a partial-recharge policy. Fortunately, adapting the model to consider a full-recharge policy is straightforward. Fig. B.1(e) illustrates the battery level function after applying the charging step. Observe that there is a prefix of the domain $P = [\min(\mathrm{dom}(\lambda_j^p)), t_2] \subseteq \mathrm{dom}(\lambda_j^p)$ where $\lambda_j^p(t) < B = 10$ for all $t \in P$. Under a full-recharge policy, it is infeasible to depart from a recharge station $j \in V_s$ with that battery level.

Formally, let p be a path ending at a station $j \in V_s$, and let λ_j^p be its battery level function at vertex j. Then, all time instants $t \in \mathrm{dom}(\lambda_j^p)$ such that $\lambda_j^p(t) < B$ must be discarded. Observe that a full-recharge policy can be enforced by setting $MBR(j) = B$ for all recharging stations $j \in V_s$ since Rule 2 removes any time from the domain under this value.

Another variant introduced by Desaulniers et al. (2016) is the possibility to restrict the number of en-route recharges. Given a maximum number of recharges per route $R_{\max}$, extend the definition of a label L of to consider a new resource $r(L)$ indicating the number of recharges performed, and forbid an extension to a recharging station if $r(L) \geq R_{\max}$. Additionally, the dominance Rule 3 must be modified by including condition $r(M) \leq r(L)$, which enforces that all extensions of L are also feasible for M.

4.4.4. Pricing heuristics

The execution of the column generation is usually accelerated by resorting to heuristics in order to identify columns with a negative reduced cost, while the exact labeling algorithm is executed if no such column is identified. One type of heuristic adaptation frequently used for labeling algorithms is to reduce the size of the graph. At the beginning of each pricing iteration, the graph is reduced to consider only the k outbound arcs $(i, j) \in A$ from each vertex $i \in V$ with the smallest reduced cost $\bar{c}_{ij}$. We refer to this approach as the k-shrink heuristic.

We further consider relaxing the dominance Rule 3 within the pricing algorithm, maintaining the feasibility of the solutions but eventually discarding routes with a negative reduced cost due to a weaker dominance criterion. We refer to the heuristic where condition (iv) is relaxed from Rule 3 as *relax-S*, while the heuristic relaxing condition (v) (i.e., the battery level) is referred to as *relax-B*.

4.4.5. Alternative objective functions

Although the TDEVRPTW minimizes the total cost of the routes, we include a brief discussion regarding the feasibility and the impact of considering further alternative objective functions. In some time-dependent problems, such as the TDTSP (see, e.g. Cordeau et al., 2014), the objective is to minimize the makespan of the route for a vehicle that departs at the beginning of the planning horizon, i.e., at time $t_0 = 0$. Our approach can be easily adapted to consider such an objective function. In this case, the reduced cost of a label L at time $t \in \mathrm{dom}(L)$ is defined as $t - \sum_{i \in p(L)} \pi_i$, where π are the values of the dual variables. Within the pricing algorithm, setting the cost $c_{ij} = 0$ for all arcs $(i, j) \in A$ redefines $c(L)$ as the sum of the dual prices of path $p(L)$ for a label L. Note that the objective function does not affect the extension rules, which remain unchanged. However, dominance rules depend on the reduced cost of the route. For a label L, when reaching $v(L)$ at some time t the reduced cost is $t - c(L)$, and thus condition (iii) from Rule 3 effectively compares the reduced cost of two labels L and M at some fixed time t. Note that even when partial dominance can discard some specific intervals, the optimal route preserves its makespan in the domain.

A more complex objective function is considered for the TDVRPTW (see, e.g. Dabia et al., 2013; Lera-Romero et al., 2020b) where vehicles are allowed to delay their departure from the depot in order to minimize the duration of the route. Unfortunately, handling this objective function within the BCP algorithm proposed for the TDEVRPTW introduces significant complexity compared to the TDVRPTW. The reason is that a new decision variable is added, which is the initial departing time is t_0. Given a path p, the battery level depends both on the completion time t and on the initial departing time t_0, and becomes a two-dimensional function

due to the delays both in the departure from the depot as well as from the departure from a vertex, as discussed in Section 2.3. Although the dominance procedure can be adapted, it would become computationally too expensive to store this type of function within each label and perform the corresponding comparisons in the dominance step. Similarly, considering energy-dependent costs for the path during the trip would suffer from the same drawback. Note, however, that a fixed recharge cost c_j can be easily included for each recharge made at station $j \in V_s$ by defining $c_{ij} := c_{ij} + c_j$ for $(i, j) \in A$.

4.5. Cutting planes

We incorporate the Subset Row Inequalities (SRCs) proposed by Jepsen et al. (2008) as cutting planes to tighten the LP relaxation, aiming to reduce the size of the branching tree. These valid inequalities have become a standard within VRP exact algorithms based on the set-partitioning formulation (see, e.g. Costa et al., 2019) and have also been used for the EVRPTW by Desaulniers et al. (2016). Formally, an SRC $s = (S, k)$ is defined by a set of customers $S \subseteq V_c$, and a coefficient $k \in \mathbb{N}$. Given a route $r \in \Omega$ and $s = (S, k)$, let $v_{sr} = \left\lfloor \frac{|S \cap r|}{k} \right\rfloor$ be the number subsets in S having k vertices that are visited by r. Then, the SRC reads

$$\sum_{r \in \Omega} y_r v_{sr} \leq \left\lfloor \frac{|S|}{k} \right\rfloor. \tag{16}$$

We consider the *3-SRC*, i.e., having $|S| = 3$ and $k = 2$. Note that since the SRCs are non-robust cuts, we refer the reader to Costa et al. (2019) for the details on how to adapt the pricing problem. Briefly, our cutting plane algorithm is as follows. After solving the LP relaxation via column generation, a maximum of n_{cuts} violated inequalities are iteratively added to the formulation following a maximum violation criterion, reoptimizing the RMP at each step without solving the pricing problem. Preliminary experiments indicate that this approach tends to reduce the number of cuts added to the RMP while preserving their impact on the quality of the LP relaxation.

5. Computational experiments

In this section, we evaluate the time-dependent model proposed in Section 2 and the BCP algorithm developed in Section 4 for the TDEVRPTW. For this purpose, we introduce a set of time-dependent benchmark instances, generated as an extension of the ones proposed by Desaulniers et al. (2016), as described in Section 5.1. The experiments are designed to obtain managerial insights regarding the importance of incorporating time-dependent travel times and, in addition, provide strong evidence about the effectiveness of our approach through extensive experiments with and without variable travel speeds. The experiments are conducted on an Intel(R) Core(TM) i7-8700 CPU @ 3.20 gigahertz with 32 gigabyte of RAM, and the algorithms are coded in C++ and use CPLEX 12.9 as an LP solver.

5.1. Experimental setup

In order to evaluate the different components of the model, we extend the instances proposed in Desaulniers et al. (2016) for the EVRPTW by incorporating time-dependent travel speeds, nonlinear recharging times, and time-dependent waiting times as follows.

Regarding time-dependent travel speeds, we we follow the strategy proposed by Dabia et al. (2013). The planning horizon $[0, T = b_d]$ is partitioned into five travel speed zones $T_1 = [0, 0.2T]$, $T_2 = [0.2T, 0.4T]$, $T_3 = [0.4T, 0.6T]$, $T_4 = [0.6T, 0.8T]$, $T_5 = [0.8T, 1.0T]$. Each arc $(i, j) \in A$ is randomly assigned to one of three speed profiles representing different traffic congestion levels: *high*, *normal*, and *low*. Fig. 5(a) shows the speed function v_p assigned to each speed profile. To enable meaningful qualitative comparisons, each speed function v_p has an average speed of 1.0 over the planning horizon, which is the travel speed considered in the time-independent instances. The parameter r required to define the instantaneous consumption function $P(v)$ is obtained from the original instances.

All instances are enhanced with piecewise linear recharging functions following the pattern proposed by Montoya et al. (2017). Recall that the original EVRPTW instances from Desaulniers et al. (2016) assume the charging of the battery is proportional to the amount of energy to be recharged, i.e. the full charge of the battery with capacity B takes t_g units of time, recharging B/t_g kWh per unit. We define the piecewise linear recharging functions where a full recharge of a battery with capacity B also requires t_g units of time as well, although the recharge per unit of time is not uniform along the interval (Fig. 5(b)). In addition, each station belongs to one of the following modes: *fast*, *medium*, and *slow*. The medium mode is defined according to B and t_g, while the slow and fast modes are a scaled version of the medium mode, $2t_g$, and $0.5t_g$, respectively.

Regarding the time-dependent waiting times at stations, Keskin et al. (2019) consider multiple scenarios that capture different real-life situations. As our objective is more general than evaluating only the waiting times, the instances are extended according to the *TD-Smooth-Long* pattern, as it represents an intermediate scenario where stations are neither too congested nor empty. Fig. 5(c) shows the function considered, which is scaled for each instance according to the planning horizon.

Note that the benchmark instances proposed by Desaulniers et al. (2016) can be retrieved by setting all the speed profiles to $v_p(t) = 1$, defining the recharging functions with a unique piece requiring the same total time as the medium charger to charge the battery completely, removing the waiting times by setting $\omega_s(t) = 0$ for every station $s \in V_s$. For the experiments, we define the following configurations for the instances:

- Basic: instances from Desaulniers et al. (2016) extended with time-dependent information.
- NC: Basic extended with non-linear charging functions.
- WT: Basic extended with time-dependent waiting time functions.
- All: Basic extended with non-linear charging and time-dependent waiting time functions.

To evaluate the impact of the time-dependent travel speeds, we further consider the time-independent versions of the above benchmarks as indicated previously.

The configurations of our BCP algorithm relies on several parameters, some of them dependent on the type of instances considered. Regarding the heuristic pricing, we execute the relax-*S*, relax-*B*, and *k*-shrink described in Section 4.4.4 with different combinations of parameters. Based on limited preliminary experiments, the configuration is as follows: relax-*S* first combined with *k*-shrink for $k = 3, 7$ and 12; relax-*B*; relax-*S*; and finally 7-shrink. Once a configuration fails, is not attempted again in future iterations, if any. If all heuristics fail to find a column with a negative reduced cost, then the exact algorithm is executed. As for the cutting planes, we set $n_{\max} = 200$.

The recharge policy considered also impacts the configuration of the BCP algorithm, in particular, whether single (S) or multiple (M) recharges are allowed, and if they are restricted to be full (F) or partial (P). Under SF and MF recharge policies, we set $MBR(j) = B$ for all $j \in V_s$. For SF and SP, we set $R_{\max} = 1$ to limit the number of recharges en route to 1.

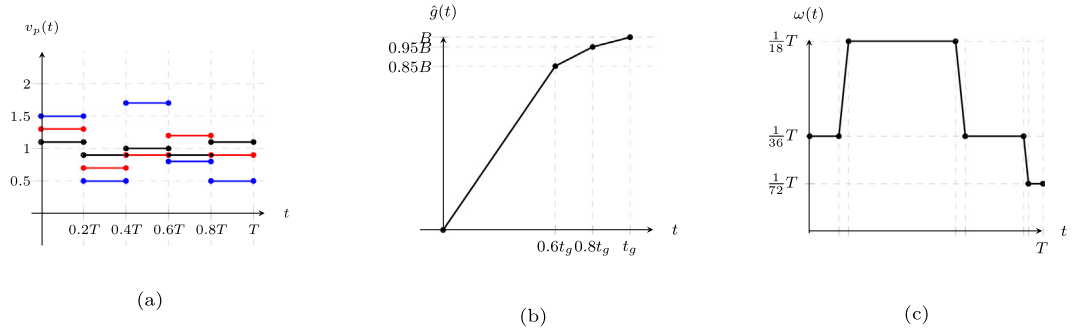

Fig. 5. Description of the dataset, including the speed function $v_p(t)$ for each profile p high (blue), normal (red), low (black) (Fig. 5(a)); a generic recharging function $\hat{g}$ from Montoya et al. (2017) (Fig. 5(b)); and waiting time function $\omega(t)$ from Keskin et al. (2019) (Fig. 5(c)). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 2
Comparison of the quality of optimal solutions: time-independent vs time-dependent.

Dataset	$\|V_c\|$	common	Time independent				Time dependent			
			cost	makespan	#rech	#routes	cost (%)	makespan (%)	#rech (%)	#routes (%)
	25	56	458.50	2879.88	1.09	4.11	−0.20	−0.05	6.82	−3.04
Basic	50	43	702.85	4441.24	0.97	6.63	0.33	0.02	24.57	−0.70
	100	14	1251.98	5449.47	1.28	12.93	0.38	2.16	4.53	2.21
	25	56	457.03	2846.69	1.10	4.02	0.01	0.72	11.92	−2.22
NC	50	45	699.23	4256.58	1.08	6.51	0.55	0.84	14.13	0.00
	100	14	1188.15	6643.29	1.21	12.14	0.61	0.50	4.07	2.94
	25	56	462.79	2948.02	0.93	4.21	0.06	−0.01	5.31	−0.85
WT	50	43	723.78	4582.24	0.89	6.88	0.46	0.91	9.68	2.36
	100	13	1224.89	7364.81	0.90	13.54	0.65	0.89	7.74	−0.57
	25	56	461.17	2906.51	0.94	4.11	0.15	0.09	5.23	−0.87
ALL	50	44	717.97	4480.35	0.88	6.91	0.69	0.58	13.98	−0.99
	100	12	1238.16	6825.26	0.90	13.17	0.48	1.75	11.35	3.16

5.2. Impact of the TDEVRPTW

The time-dependent model presented in Section 2 includes more information and, therefore, is expected to produce better-quality solutions. In this section, we evaluate and compare the solutions produced when considering time dependency to their time-independent counterparts.

To evaluate the time-dependent model, we initially conduct experiments over the four datasets described in the previous section with and without time-dependent travel speeds. For the experiments, we consider the policy MP, with multiple stops and partial recharge, as it stands as the more complex and rich scenario. We evaluate the quality of the solution in each case over four metrics: total cost; the sum of the makespan of the routes; the average number of recharges per route (#rech); and the number of routes (#routes). Table 2 reports the results, disaggregated by dataset and averaged over each instance size only over the instances that are solved to optimality in both cases. Column $|V_c|$ indicates the number of customers, column *common* shows the number of instances solved to optimality in both contexts. For the time-independent case, averages for each metric are reported in absolute terms and used as a baseline. Let z_{ti} and z_{td} be the value of a given metric for the time-independent and for the time-dependent case, respectively. For the time-dependent setup, the values are reported as the relative difference with respect to the time-independent scenario, i.e. $(z_{ti} - z_{td})/z_{ti}$.

The main message of Table 2 is that incorporating the congestion at the planning level induces differences between the optimal solutions with respect to all metrics. For instance, the number of recharges per route increases by more than a 10% on average and up to 24% for the time-dependent case, suggesting that simplifying travel speeds by averaging them impacts directly on the battery discharge model. Notice, however, that including time-dependency does not necessarily lead to higher cost solutions, as is the case of the 25 customer instances for the dataset NC. In fact, considering the congestion explicitly lead to faster routes in several cases, still remaining feasible. We further highlight the differences in the number of routes, that may become meaningful from a managerial perspective.

By construction, the time-independent instances can be interpreted as the result of averaging the travel speeds from the time-dependent ones. To highlight the congestion effects, we propose a follow-up analysis by evaluating the solutions obtained for the time-independent instances under the time-dependent context. In this sense, we assume that the time-dependent instances represent a better approximation of the real-world operations, and the infeasiblities detected are the result of neglecting the time-dependency and its effects. We focus on two different sources of infeasibilities. We say a route r is *time-infeasible* if it is infeasible even when setting the battery capacity is not restrictive, i.e. $B = \infty$. Intuitively, a time-infeasible route is infeasible either due to the violation of a time window or because its makespan exceeds the planning horizon. Similarly, we say a route r is *battery-infeasible* if all the deadlines of the time windows are removed, i.e. by setting $b_i = T$ for all $i \in V$, and r is still infeasible due to insufficient battery. These infeasibilities can be further measured and quantified. We define the *time-violation* as the worst violation of a customer's time window divided by the horizon size T, and the *battery-violation* as the extra energy required to make the route feasible divided by the battery capacity B. Finally, we count the solution as infeasible if at least one of its routes is infeasible.

Table 3 reports the results of the infeasibility tests, aggregated by n, for each dataset. Column #inst indicates the number of instances solved to optimality in the time-independent context. The rest of the columns indicate the results of the metrics described above. Column #inf-s stands for the number of infeasible solutions, and columns %inf, %inf-t, and %inf-b represent the number of infeasible, time-infeasible and battery-infeasible routes, respectively,

Table 3
Feasibility analysis of the time-independent solutions evaluated in a time-dependent context.

Dataset	$\|V_c\|$	#inst	#inf-s	%inf	%inf-t	%inf-b	%Δ-t	%Δ-b
Basic	25	56	47	50.22	30.49	23.32	1.47	0.76
	50	48	41	60.07	40.28	26.15	1.61	0.54
	100	17	17	82.16	58.92	34.05	3.61	0.88
NC	25	56	47	50.45	33.64	23.64	1.49	0.61
	50	48	42	57.34	37.20	25.94	1.76	0.60
	100	16	16	74.29	48.57	32.00	3.71	0.76
WT	25	56	46	50.43	28.21	25.21	1.57	0.67
	50	48	42	62.05	39.93	28.38	1.68	0.49
	100	18	18	87.43	54.86	41.14	3.60	1.12
ALL	25	56	48	50.44	27.19	25.88	1.54	0.61
	50	48	42	57.14	36.88	27.57	1.66	0.57
	100	15	15	74.23	47.24	36.81	3.46	1.05

over the total number of routes. %Δ-t and %Δ-b indicate the average violation of the time and battery-infeasible routes, respectively.

From the results, we note that a large number of solutions obtained from the time-independent model become infeasible when evaluated under the time-dependent model. By inspecting the solutions, we note that on average more than 60% of the routes are infeasible. Similarly to the classical TDVRP, a considerable percentage of these infeasibilities is due to a violation of the time windows originated in the variations in the travel times, between 30 and 55% of the routes according to our results. However, a very interesting result is that a significant number of infeasibilities are due to the EV running out of battery while executing the route. These percentages range between 20 and 40%, depending on the scenario considered. From a managerial standpoint, these types of infeasibilities may result in considerably expensive. Finally, we note that the average violations are rather small in both cases. This may be related to the parameters considered for the instances. Speed profiles with higher variability and a different relation between h_1 and h_2 could lead to larger values.

To further understand the practical impact of the model proposed, we extend the above experiments by considering new datasets incorporating some real-world values, aiming to incorporate at least partially some characteristics that are present in real-time networks and vehicle fleets. The datasets Basic, NC, WT, and ALL from described initially are extended into Basic-R, NC-R, WT-R, and ALL-R, respectively, as follows:

- We consdier instances with $n = 25, 50$ from the original dataset.
- The travel time information (planning horizon, travel speed, time windows, etc.) is scaled to represent more realistic travel speeds. In these new datasets, the EV travel speed ranges between 30 and 102 kilometer per hours approximately, with an average of 60 kilometer per hours. The planning horizon is expressed in hours (from 4 to 80 hours) and location distances in km (between 10 and 60 kilometers).
- Some parameters are adjusted following the ones reported in Schiffer et al. (2021). More specifically, the battery capacity and the driving range obtained when driving at the average travel speed matches the specification of a Tesla Semi 500. The information regarding clients, locations and distances is not used since, to the best of our knowledge, it is not public.

We replicated the original experiment using this new set of instances, focusing on qualitative analysis. Table 4 reports the results obtained. The key to the table is the same as in Table 2. Comparing these two tables, we observe that incorporating time-dependency into the model produces similar outcomes. The differences are consistent across all the metrics reported in relative terms on both data sets. The scaled dataset shows a smaller number of required recharges per route, likely due to the fact that the battery capacity of the Semi 500 is proportionally larger than that of the previous instances. As a result, fewer recharges per route are required and the differences across datasets are negligible for $n = 25$, and more significant for $n = 50$. In the former, waiting times and non-linear charging times have almost no impact as few recharges are performed and and the routes have few vertices, minimizing the potential impact. Finally, note that the average makespan reported for the new instances is significantly smaller because the time is measured in hours.

Table 5 replicates the qualitative analysis, using the same key as Table 3, showing some interesting insights. Note that the time-dependency remains an important factor regarding feasibility, where a significant number of routes obtained in a time-independent context become infeasible when evaluated under the time-dependent model. However, the number of infeasible routes due to battery depletion is relatively moderate. This is may be caused by different reasons. As mentioned previously, the battery capacity is a less restrictive resource in the new set of instances. Also, the vehicle information considers a mid-haul logistic context, where the weight may be relevant. In addition, our instances still assume that the impact of the load can be neglected. Note also that for larger instances that require more recharges per route, a similar behavior as that of the non-scaled dataset appears again.

5.3. The CBR branching rule

In this section, we evaluate the CBR within the framework. To reduce the potential impact of external factors, we conduct the experiments over the original time-independent instances from Desaulniers et al. (2016). Moreover, both CBR and ABR are tested on a pure BP (without the SRCs) in order better appreciate the impact when enumerating more nodes in the branch and bound tree. An important observation is that the ABR can be implemented by removing arcs from the support digraph since, for these instances, no arcs between two recharging stations exist. We consider such a simplified implementation in this experiment, although we remark that this may result beneficial for the ABR compared to the general case, given that no additional constraints must be included in the RMP and no edges are removed from the underlying graph.

Table D.7 in Appendix D presents the detailed results. The key takeout is that using CBR in this context is beneficial, as BP-CBR is able to close 15 more instances than BP-ABR. One of the reasons is the number of nodes enumerated, where CBR obtains reductions of about 50% percent on average. This effect is more evident in larger instances. However, we highlight that the average execution decreases at a smaller rate. Recall that the pricing problem needs to be modified within the CBR, and some pricing iterations may become more expensive in terms of the computation times due to this overhead, while the ABR is implemented by removing arcs.

5.4. Performance of the BCP

We conduct extensive computational experiments to evaluate the performance of the BCP algorithm under different contexts and configurations. The first experiment compares the effectiveness of the BCP on the four configurations (Basic, NC, WT, and All) for both time-dependent and time-independent versions, restricted to the MP recharge policy to maintain the number of experiments manageable. Again, we select this policy since it stands as the most challenging scenario. Table 6 reports the number of instances solved to optimality (opt), the average execution time (time), the average number of nodes enumerated (nodes), the average number of cuts added on the root node (cuts), and the average number of columns added when solving the root node (rCols) and during the entire execution (tCols). To evaluate the impact of the SRCs,

Table 4
Comparison of the quality of optimal solutions: time-independent vs. time-dependent, scaled dataset.

Dataset	$\|V_c\|$	common	Time independent				Time dependent			
			cost	makespan	#rech	#routes	cost (%)	makespan (%)	#rech (%)	#routes (%)
Basic-R	25	56	450.59	46.43	0.35	3.84	−0.07	0.60	17.51	−1.40
	50	39	698.60	73.82	0.22	6.38	0.94	1.02	88.24	0.80
NC-R	25	55	450.72	46.65	0.37	3.89	−0.28	0.50	37.22	−2.80
	50	37	696.53	75.98	0.25	6.49	0.57	0.82	51.88	1.25
WT-R	25	56	450.59	46.49	0.26	3.89	0.13	0.57	−0.07	−1.38
	50	42	690.49	71.54	0.16	6.33	48.48	−0.12	49.00	−1.50
ALL-R	25	56	450.59	46.45	0.10	3.86	0.13	0.74	145.96	−75.46
	50	45	683.69	70.08	0.18	6.38	48.65	−0.26	36.87	−73.51

Table 5
Feasibility analysis of the time-independent solutions evaluated in a time-dependent context, scaled dataset.

Dataset	$\|V_c\|$	#inst	#inf-s	%inf	%inf-t	%inf-b	%Δ-t	%Δ-b
Basic-R	25	56	31	24.53	24.53	0.00	1.74	–
	50	41	25	26.69	26.69	0.40	2.28	0.91
NC-R	25	56	31	25.00	25.00	0.00	1.77	–
	50	40	24	25.93	25.93	0.41	2.30	0.94
WT-R	25	56	31	24.19	24.19	0.00	1.72	–
	50	43	25	25.57	25.57	0.38	2.18	0.87
ALL-R	25	56	31	24.07	24.07	0.00	1.71	–
	50	45	24	23.00	23.00	0.35	2.07	0.79

Table 6
Performance for the BCP for all variants, with and without time-dependency and under a MP recharge policy.

Dataset	$\|V_c\|$	inst	Time independent									Time dependent								
			opt	time	nodes	cuts	rCols	tCols	%hg	%eg	%cg	opt	time	nodes	cuts	rCols	tCols	%hg	%eg	%cg
Basic	25	56	56	7.05	1.04	2.61	2219.00	2220.43	0.07	0.50	0.00	56	25.13	1.14	2.70	2270.75	2274.55	0.07	0.43	0.01
	50	56	48	193.27	2.88	14.79	11933.58	11999.83	0.05	0.80	0.06	44	197.67	2.00	15.23	11284.19	11471.16	0.10	0.79	0.09
	100	56	17	775.94	11.82	73.05	22141.85	24231.59	0.03	1.96	0.27	17	822.30	10.88	65.05	48783.71	58343.88	0.03	2.02	0.34
NC	25	56	56	8.72	1.25	2.29	2149.25	2158.34	0.04	0.46	0.02	56	43.16	1.11	3.25	2493.89	2498.39	0.03	0.49	0.02
	50	56	48	350.40	2.83	15.13	13961.58	14149.81	0.09	0.69	0.06	47	291.80	1.51	15.04	11045.96	11140.26	0.08	0.72	0.05
	100	56	16	536.48	9.63	63.00	52014.05	34475.38	0.03	1.91	0.22	15	446.46	7.67	58.74	36713.32	31578.73	0.03	1.64	0.19
WT	25	56	56	5.64	1.21	2.80	2106.36	2141.41	0.05	0.59	0.02	56	21.84	1.75	3.41	2162.05	2204.52	0.10	0.59	0.05
	50	56	48	219.54	6.75	16.75	10418.23	10703.42	0.07	0.95	0.10	44	165.87	3.45	15.65	9527.37	9757.80	0.08	0.90	0.12
	100	56	18	907.41	20.00	77.68	45534.16	21655.06	0.02	2.07	0.43	14	727.61	16.86	61.57	50785.70	18950.93	0.01	2.34	0.70
ALL	25	56	56	6.77	1.21	2.13	2131.71	2134.91	0.09	0.43	0.01	56	41.76	1.04	2.54	2303.34	2303.38	0.04	0.48	0.00
	50	56	48	259.72	3.75	17.13	11543.60	11720.15	0.08	0.78	0.06	47	263.20	3.21	14.83	12255.94	12517.98	0.11	0.82	0.08
	100	56	15	530.57	14.87	69.04	52582.96	22963.67	0.04	2.49	0.76	13	489.25	14.54	69.76	37435.19	19116.54	0.02	2.34	0.54

we report the average % gap (%eg) of the first LP relaxation and also at the root node (%cg). In addition, to assess for the effectiveness of the pricing heuristics, we report the % gap remaining to be closed when the exact labeling algorithm is invoked in the first LP relaxation (%hg). More specifically, we measure the difference between the LP value of the RMP when the exact labeling algorithm is called for the first time and the LP relaxation. Again, these metrics are reported in an aggregated fashion over each dataset and $|V_c|$. The columns time, nodes, and tCols are averaged over the instances solved to optimality for each metric, while the columns cuts, rCols, %hg, %eg and %cg are averaged over the instances where the root node was closed.

Table 6 suggests that incorporating the different variants has a limited impact on the performance of our algorithm. This is the expected behavior for our time-dependent framework since the time-dependent waiting times are directly encoded into the travel time functions and the non-linear charging functions are handled in the extension step of the labeling algorithm. We remark, however, that the time-dependent instances appear to be slightly harder to solve compared to their time-independent counterparts, leaving all other variables fixed. For instance, the number of columns required to execute the algorithm are in the same order regardless of the variant. The pricing problem becomes harder to solve in a time-dependent context, which is reasonable considering that the battery level function within each label has potentially more breakpoints. In fact, we observe that the BCP tends to enumerate fewer nodes in a time-dependent setup compared to its time-independent counterpart for comparable execution times. The cutting planes have a positive impact causing a dramatic gap reduction at the root node. As expected, the total number of cuts increases for as n increases. We note that the BCP produces remarkable gaps at the root node, below 1% on average, as a result of combining elementary routes with SRCs. Finally, the values reported for %hg indicate that the pricing heuristics are essential to moving forward the column generation -almost to optimality- although the exact algorithm and the cutting planes are still necessary to further reduce the gap.

5.5. Benchmark with other exact algorithms

Exact algorithms based on extended formulations outperform compact models in almost all contexts, becoming the current standard within the VRP literature. This pattern is even more pronounced in the case of TDVRPs, as the LP relaxation of the compact formulations tends to be weaker due to the presence of variable travel times. To provide further evidence regarding the effectiveness of our framework using a high standard baseline, we establish a direct comparison between our BCP and the results reported by Desaulniers et al. (2016) over the original EVRPTW instances with all four recharge policies (SF, SP, MF, MP). Besides the TDEVRPTW, this is an interesting experiment itself since the BCP algorithms incorporate different components. Briefly,

Desaulniers et al. (2016) implements a bidirectional labeling algorithm, *ng*-route relaxations, and a different cutting plane algorithm. We remark, however, that our framework includes a significant overhead to manage piecewise linear functions.

The results are shown in Table D.8 in Appendix D, where the key is the same as in the previous experiments. In this case, averages are computed over all instances solved to optimality by the corresponding method. Indeed, we are able to solve to optimality 13 and 45 more instances compared to the mono and bi-directional versions from Desaulniers et al. (2016), respectively. This indicates that our BCP is competitive, although we remark that our computer is approximately 20% faster. Besides the computation times, the remaining metrics suggest that some new components of the algorithm have a positive impact. The number of nodes enumerated remains moderate, particularly for instances with $n = 50, 100$. The root gaps share the same behavior as they are smaller than %0.20 on average. We believe this is due to a combination of factors, namely considering elementary routes that lead to tighter lower bounds, the CBR branching, and the enhanced preprocessing rules. We emphasize that improved results for the EVRPTW are reported in Desaulniers et al. (2020) using an enhanced variable fixing method.

6. Conclusions

In this paper, we propose a general model for the TDEVRPTW that accounts for the effect of congestion on the discharge of the battery. Moreover, we develop a general framework that unifies and integrates several realistic characteristics considering some of the most relevant operational constraints for the EVRP from the literature, so far addressed independently in the related literature. From a managerial standpoint, we show that neglecting the time-dependent travel speeds at the planning level can affect the quality of the routing plans, potentially incurring in violations of the time windows and exceeding the driving range. Our experiments suggest that up to 40% of the infeasibilities of the distribution, plan can be caused by exceeding the battery capacity of the EVs, and that this effect is more moderate in less stressed scenarios. From an algorithmic perspective, we develop a new BCP algorithm that generalizes existing methods for the EVRPTW, capable of handling time-dependent waiting times, non-linear charging functions, time-dependent travel times, and speed-dependent battery discharge functions. We introduce a new branching rule derived to efficiently manage the intermediate stops between customers. Overall, the proposed algorithm shows to be very effective on a large set of instances from a wide spectrum of scenarios, solving instances with up to 100 customers to optimality and with a performance comparable to the literature for the (time-independent) EVRPTW.

As future research, we believe that the evaluation of the model on real data is one of the most interesting directions. This would bring the model and the experiments closer to real-world operations. In terms of the algorithm, incorporating additional components such as variable fixing could increase the size of the instances solved. An interesting and challenging problem aiming to reduce the gap between the models and their practical implementation regards devising real-time algorithms capable of adjusting the routes during their execution to account for possible changes in the traffic conditions and other deviations from the original schedule. A promising research line regards the adaptation of the joint vehicle routing and speed optimization problem studied in Fukasawa et al. (2018) to consider routing with EVs, eventually considering a heuristic framework and the single-vehicle to re-optimize each route independently. Furthermore, integrated models exploring the impact of both the load and the speed would be challenging algorithmically but could lead to meaningful managerial insights.

Acknowledgments

This research has been founded by FONCyT grants PICT-2016-2677 and PICT-2018-2961 from the Ministry of Science, Argentina, and by the Google Latin America Research Award 2019 and 2020. The authors are grateful to the three anonymous reviewers and the Editor in Chief for their valuable comments, which have significantly improved previous versions of the paper.

Appendix A. Proof of Proposition 1

Proof. Suppose f has $|f|$ pieces $f_1, \ldots, f_{|f|}$ in the domain $[b(x), b(y)]$, and let $b(x) = u_0, \ldots, u_{|f|} = b(y)$ be its breakpoints. Since b is continuous and non-decreasing, it follows that there exist $x = t_0 < \ldots < t_{|f|} = y$ such that $b(t_i) = u_i$ for every $1 \leq i \leq |f|$. By induction on $i = 0, \ldots, |f|$ we shall prove that G is continuous, piecewise linear, and non-decreasing in the domain $[t_0, t_i]$. The base case $i = 0$ is trivial. For the inductive step $i + 1$, let $z = \max\{f(t') \mid t' \in [u_i, u_{i+1}]\}$. If $z \leq G(t_i)$, then $G(t) = G(t_i) = \max\{f(t') \mid t' \in [u_0, u_i]\}$ for every $t \in [t_i, t_{i+1}]$; clearly G is continuous, piecewise linear, and non-decreasing in $[t_0, t_{i+1}]$. Otherwise, since f is continuous and f_i is linear, it follows that f_i is increasing and there exists some $u^* \in [u_i, u_{i+1}]$ such that $f_i(u^*) = G(t_i) \geq f(u_i)$. Note, also, that $u^* = b(t^*)$ for some $t^* \in [t_i, t_{i+1}]$ because b is continuous. Therefore, $G(t) = G(t_i)$ for every $t \in [t_i, t^*]$ and, since b is non-decreasing, $G(t) = f_i(b(t)) = \max\{f(t') \mid t' \in [u_0, b(t)]\}$ for every $t \in [t^*, t_{i+1}]$. Since f_i is linear, we conclude that G is continuous, piecewise linear, and non-decreasing in $[t_0, t_{i+1}]$. □

Appendix B. Example for the battery level function

Fig. B.1 illustrates the idea of the battery level function and its different steps. Suppose there is a two-vertex path $p = (i, j)$ traveled by an EV for $i, j \in V$. For simplification purposes suppose there

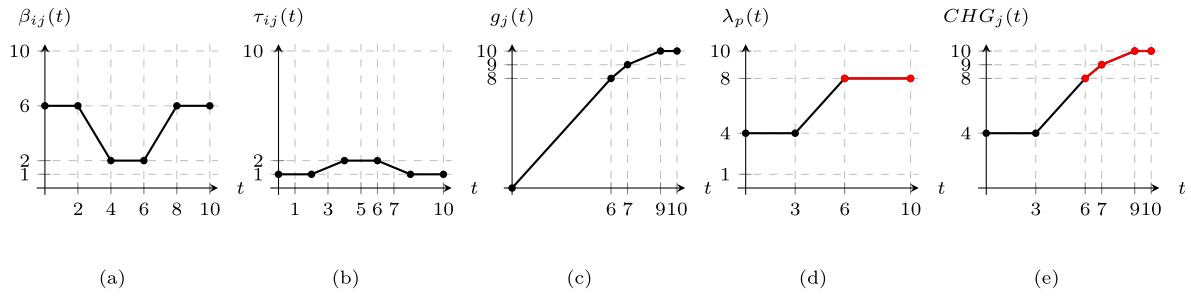

Fig. B1. Example of the battery level function $\lambda_p(t)$, including the discharge function β_{ij} (B.6a), travel time τ_{ij} (B.6b), charge function $g_j(t)$ (B.6c), TRV_{ij}, $j \in V \setminus V_s$ (B.6d), and CHG_j, $j \in V_s$ (B.6e).

Table D.7
Comparison of the different branching strategies from Section 4.3.

Dataset	$\|V_c\|$	common	CBR			ABR		
			opt	time	nodes	opt	time	nodes
	25	56	56	2.26	5.29	56	2.59	6.39
SF	50	45	51	76.10	38.11	51	89.48	55.80
	100	10	12	451.64	221.60	12	477.81	273.60
	25	55	56	7.49	5.55	55	9.89	7.76
MF	50	45	49	205.46	25.04	48	176.50	67.84
	100	10	12	509.64	89.40	11	740.99	173.20
	25	56	56	2.08	4.39	56	2.19	6.96
SP	50	48	53	139.21	42.04	50	180.41	66.50
	100	16	17	554.79	45.75	16	661.23	80.75
	25	55	56	8.39	5.22	55	11.62	12.53
MP	50	45	50	247.63	35.80	46	348.45	120.96
	100	9	9	399.76	48.11	9	727.74	92.56

Table D.8
Comparison with numerical results from Desaulniers et al. (2016) for the time-independent scenario.

Var.	$\|V_c\|$	inst	Desaulniers et al. (2016)								BCP			
			Monodirectional				Bidirectional							
			opt	time	nodes	%rg	opt	time	nodes	%rg	opt	time	nodes	%rg
	25	56	55	13.90	1.25	0.08	56	30.60	1.25	0.08	56	2.23	1.36	0.03
SF	50	56	46	265.97	7.74	0.14	47	250.87	7.47	0.16	47	92.28	2.74	0.09
	100	56	13	575.68	33.92	0.12	14	615.30	36.43	0.13	19	552.28	18.16	0.23
	25	56	55	112.98	1.22	0.04	56	16.03	1.21	0.04	56	1.78	1.11	0.01
SP	50	56	44	255.35	4.68	0.09	50	297.38	7.16	0.12	49	162.39	3.12	0.08
	100	56	18	578.65	6.44	0.06	22	628.40	15.36	0.08	22	572.31	12.64	0.12
	25	56	53	93.22	1.19	0.03	56	93.43	1.21	0.03	56	11.08	1.50	0.02
MF	50	56	44	357.85	27.09	0.05	46	258.94	31.17	0.05	46	138.58	1.74	0.04
	100	56	10	599.27	17.20	0.10	13	780.16	77.15	0.14	18	625.86	12.89	0.19
	25	56	52	151.33	1.08	0.01	54	23.48	1.11	0.01	56	7.05	1.04	0.00
MP	50	56	37	374.72	85.43	0.12	45	494.80	82.78	0.12	48	193.27	2.88	0.06
	100	56	15	855.50	5.40	0.10	18	576.32	13.11	0.11	17	775.94	11.82	0.13

are no time windows ($a_i = a_j = 0$, $b_i = b_j = T$), vertex $i = o$ is the depot, and the travel time is illustrated in Fig. B.1(b), the battery capacity is $B = 10$ and the discharge function β_{ij} is as described in Fig. B.1(a). Finally, let $\hat{g}(t)$ be the non-linear recharging function as defined in Fig. B.1(c).

The single vertex path $p_0 = (i) = (o)$ has a constant battery level function $\lambda_{p_0}(t) = B$ for all $t \in [0, T]$, which represents a full-charge when departing from the depot. Then, the battery level function λ_p results from extending λ_{p_0} over the arc (i, j). In this case, we have two possible scenarios: (i) $j \in V \setminus V_s$ or (ii) $j \in V_s$.

In the first scenario, the battery level function λ_p after traversing arc (i, j) is depicted in Fig. B.1(d). Here, the red segment shows the times when the battery level was achieved by means of waiting. For example, the battery level if arriving at j **exactly** at time $t = 10$ is 4. This occurs since vertex i must be departed from at time $t' = 9$, and then the battery consumption must be 6. Observe however that $\lambda_p(10) = 8$, since the EV can arrive at j at time $t' = 6 < 10$ with battery level 8 and then wait 4 units of time.

The second scenario shows the battery level function λ_p when $j \in V_s$ is a recharging station. In this case, function CHG_j is applied and the result is illustrated in Fig. B.1(e). The red portion of the domain includes those times when the battery level was achieved after performing a recharge.

Appendix C. Proof of Proposition 3

Proof. Clearly, having $\bar{y}_r \in \{0, 1\}$ for $r \in \Omega$ guarantees $\bar{z}_{ij} \in \{0, 1\}$ by definition since each arc (i, j) can be present in at most one route having $\bar{y}_r > 0$. Conversely, recall that for each customer set $S \subseteq V_c$ there is at most one route that visits exactly S in Ω. Additionally, constraints (12), (14) and hypothesis that $\bar{z}_{ij} \in \{0, 1\}$ enforce that for each customer $k \in V_c$ there is exactly one active variable $\bar{z}_{kj}$ and $\bar{z}_{ik}$ for some $i, j \in V_c \cup \{o, d\}$. If two active routes $r_1, r_2 \in \Omega$ such that $\bar{y}_{r_1}, \bar{y}_{r_2} > 0$ share a customer $k \in V_c$, then the customer vertex following k in $CR(r_1)$ and $CR(r_2)$ must be the same (or the previous vertex is the depot). An analogous argument also holds for the previous customer vertex. As a result, it is simple to verify that $CR(r_1) = CR(r_2)$, and then $r_1 = r_2$ since there is a unique route in Ω for each $S \subseteq V_c$. □

Appendix D. Intances and detailed results

Table D.7 shows the results for the two branching rules, CBR and ABR. We report the number of instances solved to optimality (opt), the total execution time (time), and the number of nodes enumerated (nodes) aggregated by each recharge policy and instance customer count $|V_c|$ and averaged over the instances solved by both methods (common).

The results of the comparison with the methods developed by Desaulniers et al. (2016) are shown in Table D.8, where the key is the same as in the previous experiments. In this case, averages are computed over all instances solved to optimality by the corresponding method. To enable a comparison, the values of %rg for the algorithms developed Desaulniers et al. (2016) are computed over all instances based on the results they report.

References

Adamo, T., Ghiani, G., & Guerriero, E. (2019). An enhanced lower bound for the time-dependent travelling salesman problem. *Computers and Operations Research, 113*, 104795.

Bessi, D., Ceselli, A., & Righini, G. (2022). A route-based algorithm for the electric vehicle routing problem with multiple technologies. *Technical report*.

Cordeau, J.-F., Ghiani, G., & Guerriero, E. (2014). Analysis and branch-and-cut algorithm for the time-dependent travelling salesman problem. *Transportation Science, 48*(1), 46–58.
Costa, L., Contardo, C., & Desaulniers, G. (2019). Exact branch-price-and-cut algorithms for vehicle routing. *Transportation Science, 53*(4), 946–985.
Dabia, S., Ropke, S., van Woensel, T., & De Kok, T. (2013). Branch and price for the time-dependent vehicle routing problem with time windows. *Transportation Science, 47*(3), 380–396.
Davis, B. A., & Figliozzi, M. A. (2013). A methodology to evaluate the competitiveness of electric delivery trucks. *Transportation Research Part E: Logistics and Transportation Review, 49*(1), 8–23.
Demir, E., Bektaş, T., & Laporte, G. (2014). A review of recent research on green road freight transportation. *European Journal of Operational Research, 237*(3), 775–793.
Desaulniers, G., Errico, F., Irnich, S., & Schneider, M. (2016). Exact algorithms for electric vehicle-routing problems with time windows. *Operations Research, 64*(6), 1388–1405.
Desaulniers, G., Gschwind, T., & Irnich, S. (2020). Variable fixing for two-arc sequences in branch-price-and-cut algorithms on path-based models. *Transportation Science, 54*(5), 1170–1188.
Desrosiers, J., Dumas, Y., Solomon, M. M., & Soumis, F. (1995). Time constrained routing and scheduling. In *Network routing*. In *Handbooks in operations research and management science: vol. 8* (pp. 35–139). Elsevier.
Erdoğan, S., & Miller-Hooks, E. (2012). A green vehicle routing problem. *Transportation Research Part E: Logistics and Transportation Review, 109*, 100–114.
Felipe, A., Ortuño, M. T., Righini, G., & Tirado, G. (2014). A heuristic approach for the green vehicle routing problem with multiple technologies and partial recharges. *Transportation Research Part E: Logistics and Transportation Review, 71*, 111–128.
Fetene, G. M., Kaplan, S., Mabit, S. L., Jensen, A. F., & Prato, C. G. (2017). Harnessing big data for estimating the energy consumption and driving range of electric vehicles. *Transportation Research Part D: Transport and Environment, 54*, 1–11.
Froger, A., Mendoza, J. E., Jabali, O., & Laporte, G. (2019). Improved formulations and algorithmic components for the electric vehicle routing problem with nonlinear charging functions. *Computers and Operations Research, 104*, 256–294.
Fukasawa, R., He, Q., Santos, F., & Song, Y. (2018). A joint vehicle routing and speed optimization problem. *INFORMS Journal on Computing, 30*(4), 694–709.
Gendreau, M., Ghiani, G., & Guerriero, E. (2015). Time-dependent routing problems: A review. *Computers and Operations Research, 64*, 189–197.
Goeke, D., & Schneider, M. (2015). Routing a mixed fleet of electric and conventional vehicles. *European Journal of Operational Research, 245*(1), 81–99.
Heni, H. (2018). *Optimization of time-dependent routing problems considering dynamic paths and fuel consumption*. Université Laval Ph.D. thesis..
Huang, Y., Zhao, L., Van Woensel, T., & Gross, J.-P. (2017). Time-dependent vehicle routing problem with path flexibility. *Transportation Research Part B: Methodological, 95*, 169–195.
Ichoua, S., Gendreau, M., & Potvin, J.-Y. (2003). Vehicle dispatching with time-dependent travel times. *European Journal of Operational Research, 144*(2), 379–396.
Irnich, S. (2008). Resource extension functions: Properties, inversion, and generalization to segments. *OR Spectrum, 30*(1), 113–148.
Irnich, S., & Desaulniers, G. (2005). *Shortest path problems with resource constraints*. In G. Desaulniers, J. Desrosiers, & M. M. Solomon (Eds.) (pp. 33–65)). Springer US.
Jepsen, M., Petersen, B., Spoorendonk, S., & Pisinger, D. (2008). Subset-row inequalities applied to the vehicle-routing problem with time windows. *Operations Research, 56*, 497–511.
Keskin, M., Laporte, G., & Çatay, B. (2019). Electric vehicle routing problem with time-dependent waiting times at recharging stations. *Computers and Operations Research, 107*, 77–94.
Keskin, M., & Çatay, B. (2016). Partial recharge strategies for the electric routing problem with time windows. *Transportation Research Part C: Emerging Technologies, 65*, 111–127.
Lera-Romero, G., & Miranda-Bront, J. J. (2019). A branch and cut algorithm for the time-dependent profitable tour problem with resource constraints. *European Journal of Operational Research, Icc*, 1–24.
Lera-Romero, G., Miranda-Bront, J. J., & Soulignac, F. J. (2020a). Dynamic programming for the time-dependent traveling salesman problem with time windows. *Technical report*. Optimization Online.
Lera-Romero, G., Miranda Bront, J. J., & Soulignac, F. J. (2020b). Linear edge costs and labeling algorithms: The case of the time-dependent vehicle routing problem with time windows. *Networks, 76*(1), 24–53.
Lu, J., Chen, Y., Hao, J.-K., & He, R. (2020). The time-dependent electric vehicle routing problem: Model and solution. *Expert Systems with Applications, 161*, 113593.
Montoya, A., Guéret, C., Mendoza, J. E., & Villegas, J. G. (2017). The electric vehicle routing problem with nonlinear charging function. *Transportation Research Part B: Methodological, 103*, 87–110.
Pelletier, S., Jabali, O., & Laporte, G. (2016). Goods distribution with electric vehicles: Review and research perspectives. *Transportation Science, 50*(1), 3–22.
Pelletier, S., Jabali, O., & Laporte, G. (2019). The electric vehicle routing problem with energy consumption uncertainty. *Transportation Research Part B: Methodological, 126*, 225–255.
Rastani, S., & Çatay, B. (2023). A large neighborhood search-based matheuristic for the load-dependent electric vehicle routing problem with time windows. *Annals of Operations Research, 324*, 761–793.
Restrepo, J., Rosero, J., & Tellez, S. (2014). Performance testing of electric vehicles on operating conditions in Bogot DC, Colombia. In *2014 IEEE PES transmission distribution conference and exposition – Latin America* (pp. 1–8).
Roberti, R., & Wen, M. (2016). The electric traveling salesman problem with time windows. *Transportation Research Part E: Logistics and Transportation Review, 89*, 32–52.
Sassi, O., Cherif, W. R., & Oulamara, A. (2014). Vehicle Routing Problem with Mixed fleet of conventional and heterogenous electric vehicles and time dependent charging costs. *Technical report*. https://hal.archives-ouvertes.fr/hal-01083966
Savelsbergh, M. W. P., & Van Woensel, T. (2016). 50th anniversary invited article—city logistics: Challenges and opportunities. *Transportation Science, 50*(2), 579–590.
Schiffer, M., Klein, P. S., Laporte, G., & Walther, G. (2021). Integrated planning for electric commercial vehicle fleets: A case study for retail mid-haul logistics networks. *European Journal of Operational Research, 291*(3), 944–960.
Schiffer, M., Schneider, M., Walther, G., & Laporte, G. (2019). Vehicle routing and location routing with intermediate stops: A review. *Transportation Science, 53*(2), 319–343.
Schiffer, M., & Walther, G. (2017). The electric location routing problem with time windows and partial recharging. *European Journal of Operational Research, 260*, 995–1013.
Schneider, M., Stenger, A., & Goeke, D. (2014). The electric vehicle-routing problem with time windows and recharging stations. *Transportation Science, 48*(4), 500–520.
Shao, S., Guan, W., Ran, B., He, Z., & Bi, J. (2017). Electric vehicle routing problem with charging time and variable travel time. *Mathematical Problems in Engineering, 2017*, Article ID 5098183, 13 pages.
Sun, P., Veelenturf, L. P., Dabia, S., & Van Woensel, T. (2018). The time-dependent capacitated profitable tour problem with time windows and precedence constraints. *European Journal of Operational Research, 264*(3), 1058–1073.
Tagmouti, M., Gendreau, M., & Potvin, J.-Y. (2007). Arc routing problems with time-dependent service costs. *European Journal of Operational Research, 181*(1), 30–39.
(2014). In P. Toth, & D. Vigo (Eds.), *Vehicle routing: Problems, methods, and applications* (2nd ed.). Society for Industrial and Applied Mathematics, Philadelphia, PA.
UPS (2019). Accelerating sustainable solutions, UPS 2019 sustainability progress report.
Vu, D. M., Hewitt, M., Boland, N., & Savelsbergh, M. (2020). Dynamic discretization discovery for solving the time-dependent traveling salesman problem with time windows. *Transportation Science, 54*(3), 703–720.
Wu, X., Freese, D., Cabrera, A., & Kitch, W. A. (2015). Electric vehicles' energy consumption measurement and estimation. *Transportation Research Part D: Transport and Environment, 34*, 52–67.
Zang, Y., Wang, M., & Qi, M. (2022). A column generation tailored to electric vehicle routing problem with nonlinear battery depreciation. *Computers and Operations Research, 137*, 105527.